4.10. For extensions of duality theory to problems involving general convex functions and constraint sets, see Rockafellar (1970) and Bertsekas (1995b).

4.12 Exercises 4.6 and 4.7 are adapted from Boyd and Vandenberghe (1995). The result on strict complementary slackness (Exercise 4.20) was proved by Tucker (1956). The result in Exercise 4.21 is due to Clark (1961). The result in Exercise 4.30 is due to Helly (1923). Input-output macroeconomic models of the form considered in Exercise 4.32, have been introduced by Leontief, who was awarded the 1973 Nobel prize in economics. The result in Exercise 4.41 is due to Carathéodory (1907).

Chapter 5

Sensitivity analysis

Contents

Consider the standard form problem

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

and its dual

$$\begin{aligned} \text{maximize} \quad & \mathbf{p}'\mathbf{b} \\ \text{subject to} \quad & \mathbf{p}'\mathbf{A} \leq \mathbf{c}'. \end{aligned}$$

In this chapter, we study the dependence of the optimal cost and the optimal solution on the coefficient matrix $\mathbf{A}$, the requirement vector $\mathbf{b}$, and the cost vector $\mathbf{c}$. This is an important issue in practice because we often have incomplete knowledge of the problem data and we may wish to predict the effects of certain parameter changes.

In the first section of this chapter, we develop conditions under which the optimal basis remains the same despite a change in the problem data, and we examine the consequences on the optimal cost. We also discuss how to obtain an optimal solution if we add or delete some constraints. In subsequent sections, we allow larger changes in the problem data, resulting in a new optimal basis, and we develop a global perspective of the dependence of the optimal cost on the vectors $\mathbf{b}$ and $\mathbf{c}$. The chapter ends with a brief discussion of parametric programming, which is an extension of the simplex method tailored to the case where there is a single scalar unknown parameter.

Many of the results in this chapter can be extended to cover general linear programming problems. Nevertheless, and in order to simplify the presentation, our standing assumption throughout this chapter will be that we are dealing with a standard form problem and that the rows of the $m \times n$ matrix $\mathbf{A}$ are linearly independent.

5.1 Local sensitivity analysis

In this section, we develop a methodology for performing sensitivity analysis. We consider a linear programming problem, and we assume that we already have an optimal basis $\mathbf{B}$ and the associated optimal solution $\mathbf{x}^*$. We then assume that some entry of $\mathbf{A}$, $\mathbf{b}$, or $\mathbf{c}$ has been changed, or that a new constraint is added, or that a new variable is added. We first look for conditions under which the current basis is still optimal. If these conditions are violated, we look for an algorithm that finds a new optimal solution without having to solve the new problem from scratch. We will see that the simplex method can be quite useful in this respect.

Having assumed that $\mathbf{B}$ is an optimal basis for the original problem, the following two conditions are satisfied:

$$\mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}, \qquad \text{(feasibility)}$$

$$\mathbf{c}' - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A} \geq \mathbf{0}', \qquad \text{(optimality)}.$$

When the problem is changed, we check to see how these conditions are affected. By insisting that both conditions (feasibility and optimality) hold for the modified problem, we obtain the conditions under which the basis matrix $\mathbf{B}$ remains optimal for the modified problem. In what follows, we apply this approach to several examples.

A new variable is added

Suppose that we introduce a new variable x_{n+1}, together with a corresponding column $\mathbf{A}_{n+1}$, and obtain the new problem

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x} + c_{n+1}x_{n+1} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + \mathbf{A}_{n+1}x_{n+1} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

We wish to determine whether the current basis $\mathbf{B}$ is still optimal.

We note that $(\mathbf{x}, x_{n+1}) = (\mathbf{x}^*, 0)$ is a basic feasible solution to the new problem associated with the basis $\mathbf{B}$, and we only need to examine the optimality conditions. For the basis $\mathbf{B}$ to remain optimal, it is necessary and sufficient that the reduced cost of x_{n+1} be nonnegative, that is,

$$\bar{c}_{n+1} = c_{n+1} - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}_{n+1} \geq 0.$$

If this condition is satisfied, $(\mathbf{x}^*, 0)$ is an optimal solution to the new problem. If, however, $\bar{c}_{n+1} < 0$, then $(\mathbf{x}^*, 0)$ is not necessarily optimal. In order to find an optimal solution, we add a column to the simplex tableau, associated with the new variable, and apply the primal simplex algorithm starting from the current basis $\mathbf{B}$. Typically, an optimal solution to the new problem is obtained with a small number of iterations, and this approach is usually much faster than solving the new problem from scratch.

Example 5.1 Consider the problem

$$\begin{aligned} \text{minimize} \quad & -5x_1 - x_2 + 12x_3 \\ \text{subject to} \quad & 3x_1 + 2x_2 + x_3 = 10 \\ & 5x_1 + 3x_2 + x_4 = 16 \\ & x_1, \ldots, x_4 \geq 0. \end{aligned}$$

An optimal solution to this problem is given by $\mathbf{x} = (2, 2, 0, 0)$ and the corresponding simplex tableau is given by

		x_1	x_2	x_3	x_4
	12	0	0	2	7
$x_1 =$	2	1	0	−3	2
$x_2 =$	2	0	1	5	−3

Note that $\mathbf{B}^{-1}$ is given by the last two columns of the tableau.

Let us now introduce a variable x_5 and consider the new problem

$$\begin{array}{lrcrcrcrcrcl} \text{minimize} & -5x_1 & - & x_2 & + & 12x_3 & & & - & x_5 & & \\ \text{subject to} & 3x_1 & + & 2x_2 & + & x_3 & & & + & x_5 & = & 10 \\ & 5x_1 & + & 3x_2 & & & + & x_4 & + & x_5 & = & 16 \\ & & & & & & & & & \multicolumn{3}{l}{x_1,\ldots,x_5 \geq 0.} \end{array}$$

We have $\mathbf{A}_5 = (1, 1)$ and

$$\bar{c}_5 = c_5 - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}_5 = -1 - \begin{bmatrix} -5 & -1 \end{bmatrix} \begin{bmatrix} -3 & 2 \\ 5 & -3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = -4.$$

Since $\bar{c}_5$ is negative, introducing the new variable to the basis can be beneficial. We observe that $\mathbf{B}^{-1}\mathbf{A}_5 = (-1, 2)$ and augment the tableau by introducing a column associated with x_5:

		x_1	x_2	x_3	x_4	x_5
	12	0	0	2	7	−4
$x_1 =$	2	1	0	−3	2	−1
$x_2 =$	2	0	1	5	−3	2

We then bring x_5 into the basis; x_2 exits and we obtain the following tableau, which happens to be optimal:

		x_1	x_2	x_3	x_4	x_5
	16	0	2	12	1	0
$x_1 =$	3	1	0.5	−0.5	0.5	0
$x_5 =$	1	0	0.5	2.5	−1.5	1

An optimal solution is given by $\mathbf{x} = (3, 0, 0, 0, 1)$.

A new inequality constraint is added

Let us now introduce a new constraint $\mathbf{a}_{m+1}'\mathbf{x} \geq b_{m+1}$, where $\mathbf{a}_{m+1}$ and b_{m+1} are given. If the optimal solution $\mathbf{x}^*$ to the original problem satisfies this constraint, then $\mathbf{x}^*$ is an optimal solution to the new problem as well. If the new constraint is violated, we introduce a nonnegative slack variable x_{n+1}, and rewrite the new constraint in the form $\mathbf{a}_{m+1}'\mathbf{x} - x_{n+1} = b_{m+1}$. We obtain a problem in standard form, in which the matrix $\mathbf{A}$ is replaced by

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{a}_{m+1}' & -1 \end{bmatrix}.$$

Let $\mathbf{B}$ be an optimal basis for the original problem. We form a basis for the new problem by selecting the original basic variables together with x_{n+1}. The new basis matrix $\overline{\mathbf{B}}$ is of the form

$$\overline{\mathbf{B}} = \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{a}' & -1 \end{bmatrix},$$

where the row vector $\mathbf{a}'$ contains those components of $\mathbf{a}_{m+1}'$ associated with the original basic columns. (The determinant of this matrix is the negative of the determinant of $\mathbf{B}$, hence nonzero, and we therefore have a true basis matrix.) The basic solution associated with this basis is $(\mathbf{x}^*, \mathbf{a}_{m+1}'\mathbf{x}^* - b_{m+1})$, and is infeasible because of our assumption that $\mathbf{x}^*$ violates the new constraint. Note that the new inverse basis matrix is readily available because

$$\overline{\mathbf{B}}^{-1} = \begin{bmatrix} \mathbf{B}^{-1} & \mathbf{0} \\ \mathbf{a}'\mathbf{B}^{-1} & -1 \end{bmatrix}.$$

(To see this, note that the product $\overline{\mathbf{B}}^{-1}\overline{\mathbf{B}}$ is equal to the identity matrix.)

Let $\mathbf{c}_B$ be the m-dimensional vector with the costs of the basic variables in the original problem. Then, the vector of reduced costs associated with the basis $\overline{\mathbf{B}}$ for the new problem, is given by

$$\begin{bmatrix} \mathbf{c}' & 0 \end{bmatrix} - \begin{bmatrix} \mathbf{c}_B' & 0 \end{bmatrix} \begin{bmatrix} \mathbf{B}^{-1} & \mathbf{0} \\ \mathbf{a}'\mathbf{B}^{-1} & -1 \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{a}_{m+1}' & -1 \end{bmatrix} = \begin{bmatrix} \mathbf{c}' - \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A} & 0 \end{bmatrix},$$

and is nonnegative due to the optimality of $\mathbf{B}$ for the original problem. Hence, $\overline{\mathbf{B}}$ is a dual feasible basis and we are in a position to apply the dual simplex method to the new problem. Note that an initial simplex tableau for the new problem is readily constructed. For example, we have

$$\overline{\mathbf{B}}^{-1} \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{a}_{m+1}' & -1 \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{A} & \mathbf{0} \\ \mathbf{a}'\mathbf{B}^{-1}\mathbf{A} - \mathbf{a}_{m+1}' & 1 \end{bmatrix},$$

where $\mathbf{B}^{-1}\mathbf{A}$ is available from the final simplex tableau for the original problem.

Example 5.2 Consider again the problem in Example 5.1:

$$\begin{array}{lrcrcrcrcl} \text{minimize} & -5x_1 & - & x_2 & + & 12x_3 & & & & \\ \text{subject to} & 3x_1 & + & 2x_2 & + & x_3 & & & = & 10 \\ & 5x_1 & + & 3x_2 & & & + & x_4 & = & 16 \\ & & & & & & & \multicolumn{3}{l}{x_1,\ldots,x_4 \geq 0,} \end{array}$$

and recall the optimal simplex tableau:

		x_1	x_2	x_3	x_4
	12	0	0	2	7
$x_1 =$	2	1	0	−3	2
$x_2 =$	2	0	1	5	−3

We introduce the additional constraint $x_1 + x_2 \geq 5$, which is violated by the optimal solution $\mathbf{x}^* = (2,2,0,0)$. We have $\mathbf{a}_{m+1} = (1,1,0,0)$, $b_{m+1} = 5$, and $\mathbf{a}'_{m+1}\mathbf{x}^* < b_{m+1}$. We form the standard form problem

$$\begin{array}{lrcrcrcrcrcl} \text{minimize} & -5x_1 & - & x_2 & + & 12x_3 & & & & & & \\ \text{subject to} & 3x_1 & + & 2x_2 & + & x_3 & & & & & = & 10 \\ & 5x_1 & + & 3x_2 & & & + & x_4 & & & = & 16 \\ & x_1 & + & x_2 & & & & & - & x_5 & = & 5 \\ & \multicolumn{11}{l}{x_1, \ldots, x_5 \geq 0.} \end{array}$$

Let $\mathbf{a}$ consist of the components of $\mathbf{a}_{m+1}$ associated with the basic variables. We then have $\mathbf{a} = (1,1)$ and

$$\mathbf{a}'\mathbf{B}^{-1}\mathbf{A} - \mathbf{a}'_{m+1} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 1 & 5 & -3 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 2 & -1 \end{bmatrix}.$$

The tableau for the new problem is of the form

		x_1	x_2	x_3	x_4	x_5
	12	0	0	2	7	0
$x_1 =$	2	1	0	−3	2	0
$x_2 =$	2	0	1	5	−3	0
$x_5 =$	−1	0	0	2	−1	1

We now have all the information necessary to apply the dual simplex method to the new problem.

Our discussion has been focused on the case where an inequality constraint is added to the primal problem. Suppose now that we introduce a new constraint $\mathbf{p}'\mathbf{A}_{n+1} \leq c_{n+1}$ in the dual. This is equivalent to introducing a new variable in the primal, and we are back to the case that was considered in the preceding subsection.

A new equality constraint is added

We now consider the case where the new constraint is of the form $\mathbf{a}'_{m+1}\mathbf{x} = b_{m+1}$, and we assume that this new constraint is violated by the optimal solution $\mathbf{x}^*$ to the original problem. The dual of the new problem is

$$\begin{array}{ll} \text{maximize} & \mathbf{p}'\mathbf{b} + p_{m+1}b_{m+1} \\ \text{subject to} & \begin{bmatrix} \mathbf{p}' & p_{m+1} \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{a}'_{m+1} \end{bmatrix} \leq \mathbf{c}', \end{array}$$

where p_{m+1} is a dual variable associated with the new constraint. Let $\mathbf{p}^*$ be an optimal basic feasible solution to the original dual problem. Then, $(\mathbf{p}^*, 0)$ is a feasible solution to the new dual problem.

Let m be the dimension of $\mathbf{p}$, which is the same as the original number of constraints. Since $\mathbf{p}^*$ is a basic feasible solution to the original dual problem, m of the constraints in $(\mathbf{p}^*)'\mathbf{A} \leq \mathbf{c}'$ are linearly independent and active. However, there is no guarantee that at $(\mathbf{p}^*, 0)$ we will have $m+1$ linearly independent active constraints of the new dual problem. In particular, $(\mathbf{p}^*, 0)$ need not be a basic feasible solution to the new dual problem and may not provide a convenient starting point for the dual simplex method on the new problem. While it may be possible to obtain a dual basic feasible solution by setting p_{m+1} to a suitably chosen nonzero value, we present here an alternative approach.

Let us assume, without loss of generality, that $\mathbf{a}'_{m+1}\mathbf{x}^* > b_{m+1}$. We introduce the auxiliary primal problem

$$\begin{array}{lrcrcl} \text{minimize} & \mathbf{c}'\mathbf{x} & + & Mx_{n+1} & & \\ \text{subject to} & \mathbf{A}\mathbf{x} & & & = & \mathbf{b} \\ & \mathbf{a}'_{m+1}\mathbf{x} & - & x_{n+1} & = & b_{m+1} \\ & \multicolumn{5}{l}{\mathbf{x} \geq \mathbf{0},\ x_{n+1} \geq 0,} \end{array}$$

where M is a large positive constant. A primal feasible basis for the auxiliary problem is obtained by picking the basic variables of the optimal solution to the original problem, together with the variable x_{n+1}. The resulting basis matrix is the same as the matrix $\overline{\mathbf{B}}$ of the preceding subsection. There is a difference, however. In the preceding subsection, $\overline{\mathbf{B}}$ was a dual feasible basis, whereas here $\overline{\mathbf{B}}$ is a primal feasible basis. For this reason, the primal simplex method can now be used to solve the auxiliary problem to optimality.

Suppose that an optimal solution to the auxiliary problem satisfies $x_{n+1} = 0$; this will be the case if the new problem is feasible and the coefficient M is large enough. Then, the additional constraint $\mathbf{a}'_{m+1}\mathbf{x} = b_{m+1}$ has been satisfied and we have an optimal solution to the new problem.

Changes in the requirement vector b

Suppose that some component b_i of the requirement vector $\mathbf{b}$ is changed to $b_i + \delta$. Equivalently, the vector $\mathbf{b}$ is changed to $\mathbf{b} + \delta\mathbf{e}_i$, where $\mathbf{e}_i$ is the ith unit vector. We wish to determine the range of values of δ under which the current basis remains optimal. Note that the optimality conditions are not affected by the change in $\mathbf{b}$. We therefore need to examine only the feasibility condition

$$\mathbf{B}^{-1}(\mathbf{b} + \delta\mathbf{e}_i) \geq \mathbf{0}. \tag{5.1}$$

Let $\mathbf{g} = (\beta_{1i}, \beta_{2i}, \ldots, \beta_{mi})$ be the ith column of $\mathbf{B}^{-1}$. Equation (5.1) becomes

$$\mathbf{x}_B + \delta\mathbf{g} \geq \mathbf{0},$$

or,

$$x_{B(j)} + \delta\beta_{ji} \geq 0, \qquad j = 1, \ldots, m.$$

Equivalently,

$$\max_{\{j|\beta_{ji}>0\}}\left(-\frac{x_{B(j)}}{\beta_{ji}}\right) \le \delta \le \min_{\{j|\beta_{ji}<0\}}\left(-\frac{x_{B(j)}}{\beta_{ji}}\right).$$

For δ in this range, the optimal cost, as a function of δ, is given by $\mathbf{c}_B'\mathbf{B}^{-1}(\mathbf{b}+\delta\mathbf{e}_i) = \mathbf{p}'\mathbf{b}+\delta p_i$, where $\mathbf{p}' = \mathbf{c}_B'\mathbf{B}^{-1}$ is the (optimal) dual solution associated with the current basis $\mathbf{B}$.

If δ is outside the allowed range, the current solution satisfies the optimality (or dual feasibility) conditions, but is primal infeasible. In that case, we can apply the dual simplex algorithm starting from the current basis.

Example 5.3 Consider the optimal tableau

		x_1	x_2	x_3	x_4
	12	0	0	2	7
$x_1 =$	2	1	0	-3	2
$x_2 =$	2	0	1	5	-3

from Example 5.1.

Let us contemplate adding δ to b_1. We look at the first column of $\mathbf{B}^{-1}$ which is $(-3, 5)$. The basic variables under the same basis are $x_1 = 2-3\delta$ and $2+5\delta$. This basis will remain feasible as long as $2-3\delta \ge 0$ and $2+5\delta \ge 0$, that is, if $-2/5 \le \delta \le 2/3$. The rate of change of the optimal cost per unit change of δ is given by $\mathbf{c}_B'\mathbf{B}^{-1}\mathbf{e}_1 = (-5,-1)'(-3,5) = 10$.

If δ is increased beyond $2/3$, then x_1 becomes negative. At this point, we can perform an iteration of the dual simplex method to remove x_1 from the basis, and x_3 enters the basis.

Changes in the cost vector c

Suppose now that some cost coefficient c_j becomes $c_j+\delta$. The primal feasibility condition is not affected. We therefore need to focus on the optimality condition

$$\mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A} \le \mathbf{c}'.$$

If c_j is the cost coefficient of a nonbasic variable x_j, then $\mathbf{c}_B$ does not change, and the only inequality that is affected is the one for the reduced cost of x_j; we need

$$\mathbf{c}_B'\mathbf{B}^{-1}\mathbf{A}_j \le c_j+\delta,$$

or

$$\delta \ge -\bar{c}_j.$$

If this condition holds, the current basis remains optimal; otherwise, we can apply the primal simplex method starting from the current basic feasible solution.

If c_j is the cost coefficient of the ℓth basic variable, that is, if $j = B(\ell)$, then $\mathbf{c}_B$ becomes $\mathbf{c}_B + \delta\mathbf{e}_\ell$ and all of the optimality conditions will be affected. The optimality conditions for the new problem are

$$(\mathbf{c}_B+\delta\mathbf{e}_\ell)'\mathbf{B}^{-1}\mathbf{A}_i \le c_i, \qquad \forall\, i \ne j.$$

(Since x_j is a basic variable, its reduced cost stays at zero and need not be examined.) Equivalently,

$$\delta q_{\ell i} \le \bar{c}_i, \qquad \forall\, i \ne j,$$

where $q_{\ell i}$ is the ℓth entry of $\mathbf{B}^{-1}\mathbf{A}_i$, which can be obtained from the simplex tableau. These inequalities determine the range of δ for which the same basis remains optimal.

Example 5.4 We consider once more the problem in Example 5.1 and determine the range of changes δ_i of c_i, under which the same basis remains optimal. Since x_3 and x_4 are nonbasic variables, we obtain the conditions

$$\delta_3 \ge -\bar{c}_3 = -2,$$
$$\delta_4 \ge -\bar{c}_4 = -7.$$

Consider now adding δ_1 to c_1. From the simplex tableau, we obtain $q_{12} = 0$, $q_{13} = -3$, $q_{14} = 2$, and we are led to the conditions

$$\delta_1 \ge -2/3,$$
$$\delta_1 \le 7/2.$$

Changes in a nonbasic column of A

Suppose that some entry a_{ij} in the jth column $\mathbf{A}_j$ of the matrix $\mathbf{A}$ is changed to $a_{ij}+\delta$. We wish to determine the range of values of δ for which the old optimal basis remains optimal.

If the column $\mathbf{A}_j$ is nonbasic, the basis matrix $\mathbf{B}$ does not change, and the primal feasibility condition is unaffected. Furthermore, only the reduced cost of the jth column is affected, leading to the condition

$$c_j - \mathbf{p}'(\mathbf{A}_j+\delta\mathbf{e}_i) \ge 0,$$

or,

$$\bar{c}_j - \delta p_i \ge 0,$$

where $\mathbf{p}' = \mathbf{c}_B'\mathbf{B}^{-1}$. If this condition is violated, the nonbasic column $\mathbf{A}_j$ can be brought into the basis, and we can continue with the primal simplex method.

Changes in a basic column of A

If one of the entries of a basic column $\mathbf{A}_j$ changes, then both the feasibility and optimality conditions are affected. This case is more complicated and we leave the full development for the exercises. As it turns out, the range of values of δ for which the same basis is optimal is again an interval (Exercise 5.3).

Suppose that the basic column $\mathbf{A}_j$ is changed to $\mathbf{A}_j + \delta \mathbf{e}_i$, where $\mathbf{e}_i$ is the ith unit vector. Assume that both the original problem and its dual have unique and nondegenerate optimal solutions $\mathbf{x}^*$ and $\mathbf{p}$, respectively. Let $\mathbf{x}^*(\delta)$ be an optimal solution to the modified problem, as a function of δ. It can be shown (Exercise 5.2) that for small δ we have

$$\mathbf{c}'\mathbf{x}^*(\delta) = \mathbf{c}'\mathbf{x}^* - \delta x_j^* p_i + O(\delta^2).$$

For an intuitive interpretation of this equation, let us consider the diet problem and recall that a_{ij} corresponds to the amount of the ith nutrient in the jth food. Given an optimal solution $\mathbf{x}^*$ to the original problem, an increase of a_{ij} by δ means that we are getting "for free" an additional amount δx_j^* of the ith nutrient. Since the dual variable p_i is the marginal cost per unit of the ith nutrient, we are getting for free something that is normally worth $\delta p_i x_j^*$, and this allows us to reduce our costs by that same amount.

Production planning revisited

In Section 1.2, we introduced a production planning problem that DEC had faced in the end of 1988. In this section, we answer some of the questions that we posed. Recall that there were two important choices, whether to use the constrained or the unconstrained mode of production for disk drives, and whether to use alternative memory boards. As discussed in Section 1.2, these four combinations of choices led to four different linear programming problems. We report the solution to these problems, as obtained from a linear programming package, in Table 5.1.

Table 5.1 indicates that revenues can substantially increase by using alternative memory boards, and the company should definitely do so. The decision of whether to use the constrained or the unconstrained mode of production for disk drives is less clear. In the constrained mode, the revenue is 248 million versus 213 million in the unconstrained mode. However, customer satisfaction and, therefore, future revenues might be affected, since in the constrained mode some customers will get a product different than the desired one. Moreover, these results are obtained assuming that the number of available 256K memory boards and disk drives were 8,000 and 3,000, respectively, which is the lowest value in the range that was estimated. We should therefore examine the sensitivity of the solution as the number of available 256K memory boards and disk drives increases.

Alt. boards	Mode	Revenue	x_1	x_2	x_3	x_4	x_5
no	constr.	145	0	2.5	0	0.5	2
yes	constr.	248	1.8	2	0	1	2
no	unconstr.	133	0.272	1.304	0.3	0.5	2.7
yes	unconstr.	213	1.8	1.035	0.3	0.5	2.7

Table 5.1: Optimal solutions to the four variants of the production planning problem. Revenue is in millions of dollars and the quantities x_i are in thousands.

With most linear programming packages, the output includes the values of the dual variables, as well as the range of parameter variations under which local sensitivity analysis is valid. Table 5.2 presents the values of the dual variables associated with the constraints on available disk drives and 256K memory boards. In addition, it provides the range of allowed changes on the number of disk drives and memory boards that would leave the dual variables unchanged. This information is provided for the two linear programming problems corresponding to constrained and unconstrained mode of production for disk drives, respectively, under the assumption that alternative memory boards will be used.

Mode	Constrained	Unconstrained
Revenue	248	213
Dual variable for 256K boards	15	0
Range for 256K boards	$[-1.5, 0.2]$	$[-1.62, \infty]$
Dual variable for disk drives	0	23.52
Range for disk drives	$[-0.2, 0.75]$	$[-0.91, 1.13]$

Table 5.2: Dual prices and ranges for the constraints corresponding to the availability of the number of 256K memory boards and disk drives.

In the constrained mode, increasing the number of available 256K boards by 0.2 thousand (the largest number in the allowed range) results in a revenue increase of $15 \times 0.2 = 3$ million. In the unconstrained mode, increasing the number of available 256K boards has no effect on revenues, because the dual variable is zero and the range extends upwards to infinity. In the constrained mode, increasing the number of available disk drives by up to 0.75 thousand (the largest number in the allowed range) has no effect on revenue. Finally, in the unconstrained mode, increasing the number of available disk drives by 1.13 thousand results in a revenue increase of $23.52 \times 1.13 = 26.57$ million.

In conclusion, in the constrained mode of production, it is important to aim at an increase of the number of available 256K memory boards, while in the unconstrained mode, increasing the number of disk drives is more important.

This example demonstrates that even a small linear programming problem (with five variables, in this case) can have an impact on a company's planning process. Moreover, the information provided by linear programming solvers (dual variables, ranges, etc.) can offer significant insights and can be a very useful aid to decision makers.

5.2 Global dependence on the right-hand side vector

In this section, we take a global view of the dependence of the optimal cost on the requirement vector $\mathbf{b}$.

Let

$$P(\mathbf{b}) = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$$

be the feasible set, and note that our notation makes the dependence on $\mathbf{b}$ explicit. Let

$$S = \{\mathbf{b} \mid P(\mathbf{b}) \text{ is nonempty}\},$$

and observe that

$$S = \{\mathbf{A}\mathbf{x} \mid \mathbf{x} \geq \mathbf{0}\};$$

in particular, S is a convex set. For any $\mathbf{b} \in S$, we define

$$F(\mathbf{b}) = \min_{\mathbf{x} \in P(\mathbf{b})} \mathbf{c}'\mathbf{x},$$

which is the optimal cost as a function of $\mathbf{b}$.

Throughout this section, we assume that the dual feasible set $\{\mathbf{p} \mid \mathbf{p}'\mathbf{A} \leq \mathbf{c}'\}$ is nonempty. Then, duality theory implies that the optimal primal cost $F(\mathbf{b})$ is finite for every $\mathbf{b} \in S$. Our goal is to understand the structure of the function $F(\mathbf{b})$, for $\mathbf{b} \in S$.

Let us fix a particular element $\mathbf{b}^*$ of S. Suppose that there exists a nondegenerate primal optimal basic feasible solution, and let $\mathbf{B}$ be the corresponding optimal basis matrix. The vector $\mathbf{x}_B$ of basic variables at that optimal solution is given by $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}^*$, and is positive by nondegeneracy. In addition, the vector of reduced costs is nonnegative. If we change $\mathbf{b}^*$ to $\mathbf{b}$ and if the difference $\mathbf{b} - \mathbf{b}^*$ is sufficiently small, $\mathbf{B}^{-1}\mathbf{b}$ remains positive and we still have a basic feasible solution. The reduced costs are not affected by the change from $\mathbf{b}^*$ to $\mathbf{b}$ and remain nonnegative. Therefore, $\mathbf{B}$ is an optimal basis for the new problem as well. The optimal cost $F(\mathbf{b})$ for the new problem is given by

$$F(\mathbf{b}) = \mathbf{c}_B'\mathbf{B}^{-1}\mathbf{b} = \mathbf{p}'\mathbf{b}, \qquad \text{for } \mathbf{b} \text{ close to } \mathbf{b}^*,$$

where $\mathbf{p}' = \mathbf{c}_B'\mathbf{B}^{-1}$ is the optimal solution to the dual problem. This establishes that in the vicinity of $\mathbf{b}^*$, $F(\mathbf{b})$ is a linear function of $\mathbf{b}$ and its gradient is given by $\mathbf{p}$.

We now turn to the global properties of $F(\mathbf{b})$.

Theorem 5.1 *The optimal cost $F(\mathbf{b})$ is a convex function of $\mathbf{b}$ on the set S.*

Proof. Let $\mathbf{b}^1$ and $\mathbf{b}^2$ be two elements of S. For $i = 1, 2$, let $\mathbf{x}^i$ be an optimal solution to the problem of minimizing $\mathbf{c}'\mathbf{x}$ subject to $\mathbf{x} \geq \mathbf{0}$ and $\mathbf{A}\mathbf{x} = \mathbf{b}^i$. Thus, $F(\mathbf{b}^1) = \mathbf{c}'\mathbf{x}^1$ and $F(\mathbf{b}^2) = \mathbf{c}'\mathbf{x}^2$. Fix a scalar $\lambda \in [0, 1]$, and note that the vector $\mathbf{y} = \lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2$ is nonnegative and satisfies $\mathbf{A}\mathbf{y} = \lambda\mathbf{b}^1 + (1-\lambda)\mathbf{b}^2$. In particular, $\mathbf{y}$ is a feasible solution to the linear programming problem obtained when the requirement vector $\mathbf{b}$ is set to $\lambda\mathbf{b}^1 + (1-\lambda)\mathbf{b}^2$. Therefore,

$$F\big(\lambda\mathbf{b}^1 + (1-\lambda)\mathbf{b}^2\big) \leq \mathbf{c}'\mathbf{y} = \lambda\mathbf{c}'\mathbf{x}^1 + (1-\lambda)\mathbf{c}'\mathbf{x}^2 = \lambda F(\mathbf{b}^1) + (1-\lambda)F(\mathbf{b}^2),$$

establishing the convexity of F. □

We now corroborate Theorem 5.1 by taking a different approach, involving the dual problem

$$\begin{aligned} \text{maximize} \quad & \mathbf{p}'\mathbf{b} \\ \text{subject to} \quad & \mathbf{p}'\mathbf{A} \leq \mathbf{c}', \end{aligned}$$

which has been assumed feasible. For any $\mathbf{b} \in S$, $F(\mathbf{b})$ is finite and, by strong duality, is equal to the optimal value of the dual objective. Let $\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^N$ be the extreme points of the dual feasible set. (Our standing assumption is that the matrix $\mathbf{A}$ has linearly independent rows; hence its columns span $\Re^m$. Equivalently, the rows of $\mathbf{A}'$ span $\Re^m$ and Theorem 2.6 in Section 2.5 implies that the dual feasible set must have at least one

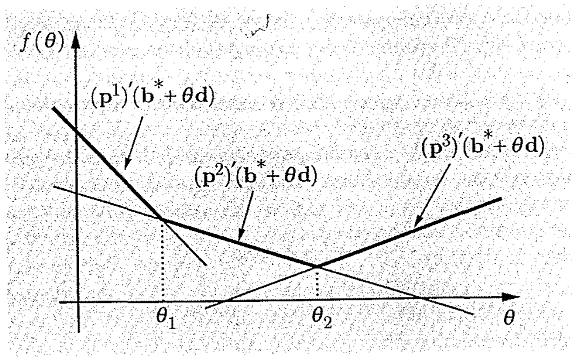

Figure 5.1: The optimal cost when the vector $\mathbf{b}$ is a function of a scalar parameter. Each linear piece is of the form $(\mathbf{p}^i)'(\mathbf{b}^* + \theta\mathbf{d})$, where $\mathbf{p}^i$ is the ith extreme point of the dual feasible set. In each one of the intervals $\theta < \theta_1$, $\theta_1 < \theta < \theta_2$, and $\theta > \theta_2$, we have different dual optimal solutions, namely, $\mathbf{p}^1$, $\mathbf{p}^2$, and $\mathbf{p}^3$, respectively. For $\theta = \theta_1$ or $\theta = \theta_2$, the dual problem has multiple optimal solutions.

extreme point.) Since the optimum of the dual must be attained at an extreme point, we obtain

$$F(\mathbf{b}) = \max_{i=1,\ldots,N} (\mathbf{p}^i)'\mathbf{b}, \qquad \mathbf{b} \in S. \tag{5.2}$$

In particular, F is equal to the maximum of a finite collection of linear functions. It is therefore a piecewise linear convex function, and we have a new proof of Theorem 5.1. In addition, within a region where F is linear, we have $F(\mathbf{b}) = (\mathbf{p}^i)'\mathbf{b}$, where $\mathbf{p}^i$ is a corresponding dual optimal solution, in agreement with our earlier discussion.

For those values of $\mathbf{b}$ for which F is not differentiable, that is, at the junction of two or more linear pieces, the dual problem does not have a unique optimal solution and this implies that every optimal basic feasible solution to the primal is degenerate. (This is because, as shown earlier in this section, the existence of a nondegenerate optimal basic feasible solution to the primal implies that F is locally linear.)

We now restrict attention to changes in $\mathbf{b}$ of a particular type, namely, $\mathbf{b} = \mathbf{b}^* + \theta\mathbf{d}$, where $\mathbf{b}^*$ and $\mathbf{d}$ are fixed vectors and θ is a scalar. Let $f(\theta) = F(\mathbf{b}^* + \theta\mathbf{d})$ be the optimal cost as a function of the scalar parameter θ. Using Eq. (5.2), we obtain

$$f(\theta) = \max_{i=1,\ldots,N} (\mathbf{p}^i)'(\mathbf{b}^* + \theta\mathbf{d}), \qquad \mathbf{b}^* + \theta\mathbf{d} \in S.$$

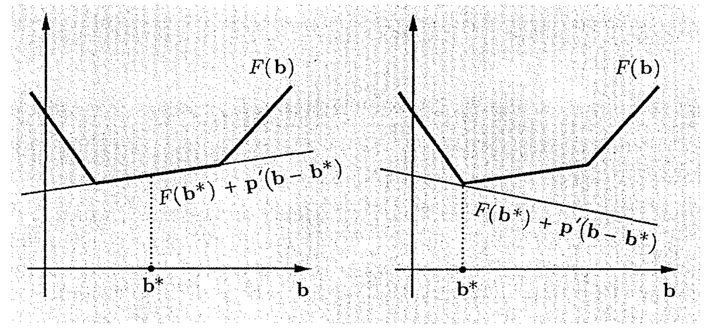

Figure 5.2: Illustration of subgradients of a function F at a point $\mathbf{b}^*$. A subgradient $\mathbf{p}$ is the gradient of a linear function $F(\mathbf{b}^*) + \mathbf{p}'(\mathbf{b} - \mathbf{b}^*)$ that lies below the function $F(\mathbf{b})$ and agrees with it for $\mathbf{b} = \mathbf{b}^*$.

This is essentially a "section" of the function F; it is again a piecewise linear convex function; see Figure 5.1. Once more, at breakpoints of this function, every optimal basic feasible solution to the primal must be degenerate.

5.3 The set of all dual optimal solutions*

We have seen that if the function F is defined, finite, and linear in the vicinity of a certain vector $\mathbf{b}^*$, then there is a unique optimal dual solution, equal to the gradient of F at that point, which leads to the interpretation of dual optimal solutions as marginal costs. We would like to extend this interpretation so that it remains valid at the breakpoints of F. This is indeed possible: we will show shortly that any dual optimal solution can be viewed as a "generalized gradient" of F. We first need the following definition, which is illustrated in Figure 5.2.

Definition 5.1 *Let F be a convex function defined on a convex set S. Let $\mathbf{b}^*$ be an element of S. We say that a vector $\mathbf{p}$ is a* **subgradient** *of F at $\mathbf{b}^*$ if*

$$F(\mathbf{b}^*) + \mathbf{p}'(\mathbf{b} - \mathbf{b}^*) \le F(\mathbf{b}), \qquad \forall\, \mathbf{b} \in S.$$

Note that if $\mathbf{b}^*$ is a breakpoint of the function F, then there are several subgradients. On the other hand, if F is linear near $\mathbf{b}^*$, there is a unique subgradient, equal to the gradient of F.

Theorem 5.2 *Suppose that the linear programming problem of minimizing $\mathbf{c}'\mathbf{x}$ subject to $\mathbf{Ax} = \mathbf{b}^*$ and $\mathbf{x} \geq \mathbf{0}$ is feasible and that the optimal cost is finite. Then, a vector $\mathbf{p}$ is an optimal solution to the dual problem if and only if it is a subgradient of the optimal cost function F at the point $\mathbf{b}^*$.*

Proof. Recall that the function F is defined on the set S, which is the set of vectors $\mathbf{b}$ for which the set $P(\mathbf{b})$ of feasible solutions to the primal problem is nonempty. Suppose that $\mathbf{p}$ is an optimal solution to the dual problem. Then, strong duality implies that $\mathbf{p}'\mathbf{b}^* = F(\mathbf{b}^*)$. Consider now some arbitrary $\mathbf{b} \in S$. For any feasible solution $\mathbf{x} \in P(\mathbf{b})$, weak duality yields $\mathbf{p}'\mathbf{b} \leq \mathbf{c}'\mathbf{x}$. Taking the minimum over all $\mathbf{x} \in P(\mathbf{b})$, we obtain $\mathbf{p}'\mathbf{b} \leq F(\mathbf{b})$. Hence, $\mathbf{p}'\mathbf{b} - \mathbf{p}'\mathbf{b}^* \leq F(\mathbf{b}) - F(\mathbf{b}^*)$, and we conclude that $\mathbf{p}$ is a subgradient of F at $\mathbf{b}^*$.

We now prove the converse. Let $\mathbf{p}$ be a subgradient of F at $\mathbf{b}^*$; that is,

$$F(\mathbf{b}^*) + \mathbf{p}'(\mathbf{b} - \mathbf{b}^*) \leq F(\mathbf{b}), \qquad \forall\, \mathbf{b} \in S. \tag{5.3}$$

Pick some $\mathbf{x} \geq \mathbf{0}$, let $\mathbf{b} = \mathbf{Ax}$, and note that $\mathbf{x} \in P(\mathbf{b})$. In particular, $F(\mathbf{b}) \leq \mathbf{c}'\mathbf{x}$. Using Eq. (5.3), we obtain

$$\mathbf{p}'\mathbf{Ax} = \mathbf{p}'\mathbf{b} \leq F(\mathbf{b}) - F(\mathbf{b}^*) + \mathbf{p}'\mathbf{b}^* \leq \mathbf{c}'\mathbf{x} - F(\mathbf{b}^*) + \mathbf{p}'\mathbf{b}^*.$$

Since this is true for all $\mathbf{x} \geq \mathbf{0}$, we must have $\mathbf{p}'\mathbf{A} \leq \mathbf{c}'$, which shows that $\mathbf{p}$ is a dual feasible solution. Also, by letting $\mathbf{x} = \mathbf{0}$, we obtain $F(\mathbf{b}^*) \leq \mathbf{p}'\mathbf{b}^*$. Using weak duality, every dual feasible solution $\mathbf{q}$ must satisfy $\mathbf{q}'\mathbf{b}^* \leq F(\mathbf{b}^*) \leq \mathbf{p}'\mathbf{b}^*$, which shows that $\mathbf{p}$ is a dual optimal solution. □

5.4 Global dependence on the cost vector

In the last two sections, we fixed the matrix $\mathbf{A}$ and the vector $\mathbf{c}$, and we considered the effect of changing the vector $\mathbf{b}$. The key to our development was the fact that the set of dual feasible solutions remains the same as $\mathbf{b}$ varies. In this section, we study the case where $\mathbf{A}$ and $\mathbf{b}$ are fixed, but the vector $\mathbf{c}$ varies. In this case, the primal feasible set remains unaffected; our standing assumption will be that it is nonempty.

We define the dual feasible set

$$Q(\mathbf{c}) = \{\mathbf{p} \mid \mathbf{p}'\mathbf{A} \leq \mathbf{c}\},$$

and let

$$T = \{\mathbf{c} \mid Q(\mathbf{c}) \text{ is nonempty}\}.$$

If $\mathbf{c}^1 \in T$ and $\mathbf{c}^2 \in T$, then there exist $\mathbf{p}^1$ and $\mathbf{p}^2$ such that $(\mathbf{p}^1)'\mathbf{A} \leq \mathbf{c}'$ and $(\mathbf{p}^2)'\mathbf{A} \leq \mathbf{c}'$. For any scalar $\lambda \in [0, 1]$, we have

$$\big(\lambda(\mathbf{p}^1)' + (1-\lambda)(\mathbf{p}^2)'\big)\mathbf{A} \leq \lambda\mathbf{c}^1 + (1-\lambda)\mathbf{c}^2,$$

and this establishes that $\lambda\mathbf{c}^1 + (1-\lambda)\mathbf{c}^2 \in T$. We have therefore shown that T is a convex set.

If $\mathbf{c} \notin T$, the infeasibility of the dual problem implies that the optimal primal cost is $-\infty$. On the other hand, if $\mathbf{c} \in T$, the optimal primal cost must be finite. Thus, the optimal primal cost, which we will denote by $G(\mathbf{c})$, is finite if and only if $\mathbf{c} \in T$.

Let $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N$ be the basic feasible solutions in the primal feasible set; clearly, these do not depend on $\mathbf{c}$. Since an optimal solution to a standard form problem can always be found at an extreme point, we have

$$G(\mathbf{c}) = \min_{i=1,\ldots,N} \mathbf{c}'\mathbf{x}^i.$$

Thus, $G(\mathbf{c})$ is the minimum of a finite collection of linear functions and is a piecewise linear concave function. If for some value $\mathbf{c}^*$ of $\mathbf{c}$, the primal has a unique optimal solution $\mathbf{x}^i$, we have $(\mathbf{c}^*)'\mathbf{x}^i < (\mathbf{c}^*)'\mathbf{x}^j$, for all $j \neq i$. For $\mathbf{c}$ very close to $\mathbf{c}^*$, the inequalities $\mathbf{c}'\mathbf{x}^i < \mathbf{c}'\mathbf{x}^j$, $j \neq i$, continue to hold, implying that $\mathbf{x}^i$ is still a unique primal optimal solution with cost $\mathbf{c}'\mathbf{x}^i$. We conclude that, locally, $G(\mathbf{c}) = \mathbf{c}'\mathbf{x}^i$. On the other hand, at those values of $\mathbf{c}$ that lead to multiple primal optimal solutions, the function G has a breakpoint.

We summarize the main points of the preceding discussion.

Theorem 5.3 *Consider a feasible linear programming problem in standard form.*

(a) *The set T of all $\mathbf{c}$ for which the optimal cost is finite, is convex.*

(b) *The optimal cost $G(\mathbf{c})$ is a concave function of $\mathbf{c}$ on the set T.*

(c) *If for some value of $\mathbf{c}$ the primal problem has a unique optimal solution $\mathbf{x}^*$, then G is linear in the vicinity of $\mathbf{c}$ and its gradient is equal to $\mathbf{x}^*$.*

5.5 Parametric programming

Let us fix $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$, and a vector $\mathbf{d}$ of the same dimension as $\mathbf{c}$. For any scalar θ, we consider the problem

$$\begin{array}{ll} \text{minimize} & (\mathbf{c} + \theta\mathbf{d})'\mathbf{x} \\ \text{subject to} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{array}$$

and let $g(\theta)$ be the optimal cost as a function of θ. Naturally, we assume that the feasible set is nonempty. For those values of θ for which the optimal cost is finite, we have

$$g(\theta) = \min_{i=1,\ldots,N} (\mathbf{c} + \theta\mathbf{d})'\mathbf{x}^i,$$

where $\mathbf{x}^1, \ldots, \mathbf{x}^N$ are the extreme points of the feasible set; see Figure 5.3. In particular, $g(\theta)$ is a piecewise linear and concave function of the parameter θ. In this section, we discuss a systematic procedure, based on the simplex method, for obtaining $g(\theta)$ for all values of θ. We start with an example.

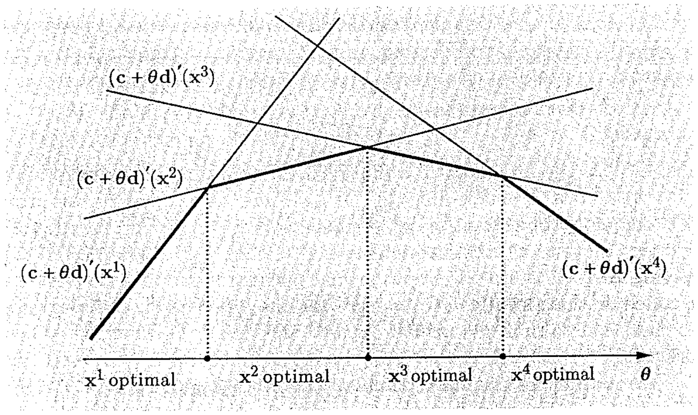

Figure 5.3: The optimal cost $g(\theta)$ as a function of θ.

Example 5.5 Consider the problem

$$\begin{array}{ll} \text{minimize} & (-3+2\theta)x_1 + (3-\theta)x_2 + x_3 \\ \text{subject to} & x_1 + 2x_2 - 3x_3 \le 5 \\ & 2x_1 + x_2 - 4x_3 \le 7 \\ & x_1, x_2, x_3 \ge 0. \end{array}$$

We introduce slack variables in order to bring the problem into standard form, and then let the slack variables be the basic variables. This determines a basic feasible solution and leads to the following tableau.

		x_1	x_2	x_3	x_4	x_5
	0	$-3+2\theta$	$3-\theta$	1	0	0
$x_4 =$	5	1	2	-3	1	0
$x_5 =$	7	2	1	-4	0	1

If $-3+2\theta \ge 0$ and $3-\theta \ge 0$, all reduced costs are nonnegative and we have an optimal basic feasible solution. In particular,

$$g(\theta) = 0, \qquad \text{if } \frac{3}{2} \le \theta \le 3.$$

If θ is increased slightly above 3, the reduced cost of x_2 becomes negative and we no longer have an optimal basic feasible solution. We let x_2 enter the basis, x_4 exits, and we obtain the new tableau:

		x_1	x_2	x_3	x_4	x_5
	$-7.5+2.5\theta$	$-4.5+2.5\theta$	0	$5.5-1.5\theta$	$-1.5+0.5\theta$	0
$x_2 =$	2.5	0.5	1	-1.5	0.5	0
$x_5 =$	4.5	1.5	0	-2.5	-0.5	1

We note that all reduced costs are nonnegative if and only if $3 \le \theta \le 5.5/1.5$. The optimal cost for that range of values of θ is

$$g(\theta) = 7.5 - 2.5\theta, \qquad \text{if } 3 \le \theta \le \frac{5.5}{1.5}.$$

If θ is increased beyond $5.5/1.5$, the reduced cost of x_3 becomes negative. If we attempt to bring x_3 into the basis, we cannot find a positive pivot element in the third column of the tableau, and the problem is unbounded, with $g(\theta) = -\infty$.

Let us now go back to the original tableau and suppose that θ is decreased to a value slightly below $3/2$. Then, the reduced cost of x_1 becomes negative, we let x_1 enter the basis, and x_5 exits. The new tableau is:

		x_1	x_2	x_3	x_4	x_5
	$10.5-7\theta$	0	$4.5-2\theta$	$-5+4\theta$	0	$1.5-\theta$
$x_4 =$	1.5	0	1.5	-1	1	-0.5
$x_1 =$	3.5	1	0.5	-2	0	0.5

We note that all of the reduced costs are nonnegative if and only if $5/4 \le \theta \le 3/2$. For these values of θ, we have an optimal solution, with an optimal cost of

$$g(\theta) = -10.5 + 7\theta, \qquad \text{if } \frac{5}{4} \le \theta \le \frac{3}{2}.$$

Finally, for $\theta < 5/4$, the reduced cost of x_3 is negative, but the optimal cost is equal to $-\infty$, because all entries in the third column of the tableau are negative. We plot the optimal cost in Figure 5.4.

We now generalize the steps in the preceding example, in order to obtain a broader methodology. The key observation is that once a basis is fixed, the reduced costs are affine (linear plus a constant) functions of θ. Then, if we require that all reduced costs be nonnegative, we force θ to belong to some interval. (The interval could be empty but if it is nonempty, its endpoints are also included.) We conclude that for any given basis, the set of θ for which this basis is optimal is a closed interval.

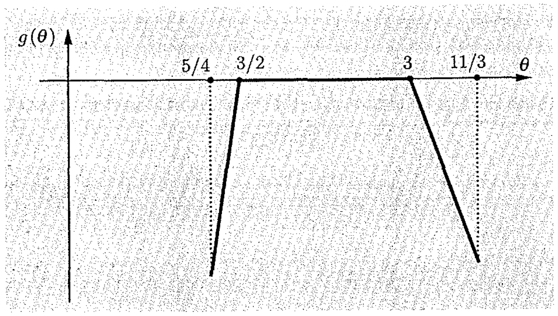

Figure 5.4: The optimal cost $g(\theta)$ as a function of θ, in Example 5.5. For θ outside the interval $[5/4, 11/3]$, $g(\theta)$ is equal to $-\infty$.

Let us now assume that we have chosen a basic feasible solution and an associated basis matrix $\mathbf{B}$, and suppose that this basis is optimal for θ satisfying $\theta_1 \leq \theta \leq \theta_2$. Let x_j be a variable whose reduced cost becomes negative for $\theta > \theta_2$. Since this reduced cost is nonnegative for $\theta_1 \leq \theta \leq \theta_2$, it must be equal to zero when $\theta = \theta_2$. We now attempt to bring x_j into the basis and consider separately the different cases that may arise.

Suppose that no entry of the jth column $\mathbf{B}^{-1}\mathbf{A}_j$ of the simplex tableau is positive. For $\theta > \theta_2$, the reduced cost of x_j is negative, and this implies that the optimal cost is $-\infty$ in that range.

If the jth column of the tableau has at least one positive element, we carry out a change of basis and obtain a new basis matrix $\overline{\mathbf{B}}$. For $\theta = \theta_2$, the reduced cost of the entering variable is zero and, therefore, the cost associated with the new basis is the same as the cost associated with the old basis. Since the old basis was optimal for $\theta = \theta_2$, the same must be true for the new basis. On the other hand, for $\theta < \theta_2$, the entering variable x_j had a positive reduced cost. According to the pivoting mechanics, and for $\theta < \theta_2$, a negative multiple of the pivot row is added to the pivot row, and this makes the reduced cost of the exiting variable negative. This implies that the new basis cannot be optimal for $\theta < \theta_2$. We conclude that the range of values of θ for which the new basis is optimal is of the form $\theta_2 \leq \theta \leq \theta_3$, for some θ_3. By continuing similarly, we obtain a sequence of bases, with the ith basis being optimal for $\theta_i \leq \theta \leq \theta_{i+1}$.

Note that a basis which is optimal for $\theta \in [\theta_i, \theta_{i+1}]$ cannot be optimal for values of θ greater than θ_{i+1}. Thus, if $\theta_{i+1} > \theta_i$ for all i, the same basis cannot be encountered more than once and the entire range of values of θ will be traced in a finite number of iterations, with each iteration leading to a new breakpoint of the optimal cost function $g(\theta)$. (The number of breakpoints may increase exponentially with the dimension of the problem.)

The situation is more complicated if for some basis we have $\theta_i = \theta_{i+1}$. In this case, it is possible that the algorithm keeps cycling between a finite number of different bases, all of which are optimal only for $\theta = \theta_i = \theta_{i+1}$. Such cycling can only happen in the presence of degeneracy in the primal problem (Exercise 5.17), but can be avoided if an appropriate anticycling rule is followed. In conclusion, the procedure we have outlined, together with an anticycling rule, partitions the range of possible values of θ into consecutive intervals and, for each interval, provides us with an optimal basis and the optimal cost function as a function of θ.

There is another variant of parametric programming that can be used when $\mathbf{c}$ is kept fixed but $\mathbf{b}$ is replaced by $\mathbf{b} + \theta\mathbf{d}$, where $\mathbf{d}$ is a given vector and θ is a scalar. In this case, the zeroth column of the tableau depends on θ. Whenever θ reaches a value at which some basic variable becomes negative, we apply the dual simplex method in order to recover primal feasibility.

5.6 Summary

In this chapter, we have studied the dependence of optimal solutions and of the optimal cost on the problem data, that is, on the entries of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$. For many of the cases that we have examined, a common methodology was used. Subsequent to a change in the problem data, we first examine its effects on the feasibility and optimality conditions. If we wish the same basis to remain optimal, this leads us to certain limitations on the magnitude of the changes in the problem data. For larger changes, we no longer have an optimal basis and some remedial action (involving the primal or dual simplex method) is typically needed.

We close with a summary of our main results.

(a) If a new variable is added, we check its reduced cost and if it is negative, we add a new column to the tableau and proceed from there.

(b) If a new constraint is added, we check whether it is violated and if so, we form an auxiliary problem and its tableau, and proceed from there.

(c) If an entry of $\mathbf{b}$ or $\mathbf{c}$ is changed by δ, we obtain an interval of values of δ for which the same basis remains optimal.

(d) If an entry of $\mathbf{A}$ is changed by δ, a similar analysis is possible. However, this case is somewhat complicated if the change affects an entry of a basic column.

(e) Assuming that the dual problem is feasible, the optimal cost is a piecewise linear convex function of the vector $\mathbf{b}$ (for those $\mathbf{b}$ for which the primal is feasible). Furthermore, subgradients of the optimal cost function correspond to optimal solutions to the dual problem.

(f) Assuming that the primal problem is feasible, the optimal cost is a piecewise linear concave function of the vector $\mathbf{c}$ (for those $\mathbf{c}$ for which the primal has finite cost).

(g) If the cost vector is an affine function of a scalar parameter θ, there is a systematic procedure (parametric programming) for solving the problem for all values of θ. A similar procedure is possible if the vector $\mathbf{b}$ is an affine function of a scalar parameter.

5.7 Exercises

Exercise 5.1 Consider the same problem as in Example 5.1, for which we already have an optimal basis. Let us introduce the additional constraint $x_1 + x_2 = 3$. Form the auxiliary problem described in the text, and solve it using the primal simplex method. Whenever the "large" constant M is compared to another number, M should be treated as being the larger one.

Exercise 5.2 (Sensitivity with respect to changes in a basic column of A) In this problem (and the next two) we study the change in the value of the optimal cost when an entry of the matrix $\mathbf{A}$ is perturbed by a small amount. We consider a linear programming problem in standard form, under the usual assumption that $\mathbf{A}$ has linearly independent rows. Suppose that we have an optimal basis $\mathbf{B}$ that leads to a nondegenerate optimal solution $\mathbf{x}^*$, and a nondegenerate dual optimal solution $\mathbf{p}$. We assume that the first column is basic. We will now change the first entry of $\mathbf{A}_1$ from a_{11} to $a_{11} + \delta$, where δ is a small scalar. Let $\mathbf{E}$ be a matrix of dimensions $m \times m$ (where m is the number of rows of $\mathbf{A}$), whose entries are all zero except for the top left entry e_{11}, which is equal to 1.

(a) Show that if δ is small enough, $\mathbf{B}+\delta\mathbf{E}$ is a basis matrix for the new problem.

(b) Show that under the basis $\mathbf{B} + \delta\mathbf{E}$, the vector $\mathbf{x}_B$ of basic variables in the new problem is equal to $(\mathbf{I} + \delta\mathbf{B}^{-1}\mathbf{E})^{-1}\mathbf{B}^{-1}\mathbf{b}$.

(c) Show that if δ is sufficiently small, $\mathbf{B} + \delta\mathbf{E}$ is an optimal basis for the new problem.

(d) We use the symbol $\approx$ to denote equality when second order terms in δ are ignored. The following approximation is known to be true: $(\mathbf{I}+\delta\mathbf{B}^{-1}\mathbf{E})^{-1} \approx \mathbf{I} - \delta\mathbf{B}^{-1}\mathbf{E}$. Using this approximation, show that

$$\mathbf{c}'_B\mathbf{x}_B \approx \mathbf{c}'\mathbf{x}^* - \delta p_1 x_1^*,$$

where x_1^* (respectively, p_1) is the first component of the optimal solution to the original primal (respectively, dual) problem, and $\mathbf{x}_B$ has been defined in part (b).

Exercise 5.3 (Sensitivity with respect to changes in a basic column of A) Consider a linear programming problem in standard form under the usual assumption that the rows of the matrix $\mathbf{A}$ are linearly independent. Suppose that the columns $\mathbf{A}_1, \ldots, \mathbf{A}_m$ form an optimal basis. Let $\mathbf{A}_0$ be some vector and suppose that we change $\mathbf{A}_1$ to $\mathbf{A}_1 + \delta\mathbf{A}_0$. Consider the matrix $\mathbf{B}(\delta)$ consisting of the columns $\mathbf{A}_0 + \delta\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m$. Let $[\delta_1, \delta_2]$ be a closed interval of values of δ that contains zero and in which the determinant of $\mathbf{B}(\delta)$ is nonzero. Show that the subset of $[\delta_1, \delta_2]$ for which $\mathbf{B}(\delta)$ is an optimal basis is also a closed interval.

Exercise 5.4 Consider the problem in Example 5.1, with a_{11} changed from 3 to $3 + \delta$. Let us keep x_1 and x_2 as the basic variables and let $\mathbf{B}(\delta)$ be the corresponding basis matrix, as a function of δ.

(a) Compute $\mathbf{B}(\delta)^{-1}\mathbf{b}$. For which values of δ is $\mathbf{B}(\delta)$ a feasible basis?

(b) Compute $\mathbf{c}'_B\mathbf{B}(\delta)^{-1}$. For which values of δ is $\mathbf{B}(\delta)$ an optimal basis?

(c) Determine the optimal cost, as a function of δ, when δ is restricted to those values for which $\mathbf{B}(\delta)$ is an optimal basis matrix.

Exercise 5.5 While solving a standard form linear programming problem using the simplex method, we arrive at the following tableau:

		x_1	x_2	x_3	x_4	x_5
		0	0	$\bar{c}_3$	0	$\bar{c}_5$
$x_2 =$	1	0	1	-1	0	β
$x_4 =$	2	0	0	2	1	γ
$x_1 =$	3	1	0	4	0	δ

Suppose also that the last three columns of the matrix $\mathbf{A}$ form an identity matrix.

(a) Give necessary and sufficient conditions for the basis described by this tableau to be optimal (in terms of the coefficients in the tableau).

(b) Assume that this basis is optimal and that $\bar{c}_3 = 0$. Find an optimal basic feasible solution, other than the one described by this tableau.

(c) Suppose that $\gamma \geq 0$. Show that there exists an optimal basic feasible solution, regardless of the values of $\bar{c}_3$ and $\bar{c}_5$.

(d) Assume that the basis associated with this tableau is optimal. Suppose also that b_1 in the original problem is replaced by $b_1 + \epsilon$. Give upper and lower bounds on ϵ so that this basis remains optimal.

(e) Assume that the basis associated with this tableau is optimal. Suppose also that c_1 in the original problem is replaced by $c_1 + \epsilon$. Give upper and lower bounds on ϵ so that this basis remains optimal.

Exercise 5.6 Company A has agreed to supply the following quantities of special lamps to Company B during the next 4 months:

Month	January	February	March	April
Units	150	160	225	180

Company A can produce a maximum of 160 lamps per month at a cost of $35 per unit. Additional lamps can be purchased from Company C at a cost of $50 per lamp. Company A incurs an inventory holding cost of $5 per month for each lamp held in inventory.

(a) Formulate the problem that Company A is facing as a linear programming problem.

(b) Solve the problem using a linear programming package.

(c) Company A is considering some preventive maintenance during one of the first three months. If maintenance is scheduled for January, the company can manufacture only 151 units (instead of 160); similarly, the maximum possible production if maintenance is scheduled for February or March is 153 and 155 units, respectively. What maintenance schedule would you recommend and why?

(d) Company D has offered to supply up to 50 lamps (total) to Company A during either January, February or March. Company D charges $45 per lamp. Should Company A buy lamps from Company D? If yes, when and how many lamps should Company A purchase, and what is the impact of this decision on the total cost?

(e) Company C has offered to lower the price of units supplied to Company A during February. What is the maximum decrease that would make this offer attractive to Company A?

(f) Because of anticipated increases in interest rates, the holding cost per lamp is expected to increase to $8 per unit in February. How does this change affect the total cost and the optimal solution?

(g) Company B has just informed Company A that it requires only 90 units in January (instead of 150 requested previously). Calculate upper and lower bounds on the impact of this order on the optimal cost using information from the optimal solution to the original problem.

Exercise 5.7 A paper company manufactures three basic products: pads of paper, 5-packs of paper, and 20-packs of paper. The pad of paper consists of a single pad of 25 sheets of lined paper. The 5-pack consists of 5 pads of paper, together with a small notebook. The 20-pack of paper consists of 20 pads of paper, together with a large notebook. The small and large notebooks are not sold separately.

Production of each pad of paper requires 1 minute of paper-machine time, 1 minute of supervisory time, and \$.10 in direct costs. Production of each small notebook takes 2 minutes of paper-machine time, 45 seconds of supervisory time, and \$.20 in direct cost. Production of each large notebook takes 3 minutes of paper machine time, 30 seconds of supervisory time and \$.30 in direct costs. To package the 5-pack takes 1 minute of packager's time and 1 minute of supervisory time. To package the 20-pack takes 3 minutes of packager's time and 2 minutes of supervisory time. The amounts of available paper-machine time, supervisory time, and packager's time are constants b_1, b_2, b_3, respectively. Any of the three products can be sold to retailers in any quantity at the prices \$.30, \$1.60, and \$7.00, respectively.

Provide a linear programming formulation of the problem of determining an optimal mix of the three products. (You may ignore the constraint that only integer quantities can be produced.) Try to formulate the problem in such a way that the following questions can be answered by looking at a single dual variable or reduced cost in the final tableau. Also, for each question, give a brief explanation of why it can be answered by looking at just one dual price or reduced cost.

(a) What is the marginal value of an extra unit of supervisory time?

(b) What is the lowest price at which it is worthwhile to produce single pads of paper for sale?

(c) Suppose that part-time supervisors can be hired at $8 per hour. Is it worthwhile to hire any?

(d) Suppose that the direct cost of producing pads of paper increases from \$.10 to \$.12. What is the profit decrease?

Exercise 5.8 A pottery manufacturer can make four different types of dining room service sets: JJP English, Currier, Primrose, and Bluetail. Furthermore, Primrose can be made by two different methods. Each set uses clay, enamel, dry room time, and kiln time, and results in a profit shown in Table 5.3. (Here, lbs is the abbreviation for pounds).

Resources	E	C	P_1	P_2	B	Total
Clay (lbs)	10	15	10	10	20	130
Enamel (lbs)	1	2	2	1	1	13
Dry room (hours)	3	1	6	6	3	45
Kiln (hours)	2	4	2	5	3	23
Profit	51	102	66	66	89	

Table 5.3: The rightmost column in the table gives the manufacturer's resource availability for the remainder of the week. Notice that Primrose can be made by two different methods. They both use the same amount of clay (10 lbs.) and dry room time (6 hours). But the second method uses one pound less of enamel and three more hours in the kiln.

The manufacturer is currently committed to making the same amount of Primrose using methods 1 and 2. The formulation of the profit maximization problem is given below. The decision variables E, C, P_1, P_2, B are the number of sets of type English, Currier, Primrose Method 1, Primrose Method 2, and Bluetail, respectively. We assume, for the purposes of this problem, that the number of sets of each type can be fractional.

$$\begin{array}{lrrrrrl} \text{maximize} & 51E + & 102C + & 66P_1 + & 66P_2 + & 89B & \\ \text{subject to} & 10E + & 15C + & 10P_1 + & 10P_2 + & 20B & \leq 130 \\ & E + & 2C + & 2P_1 + & P_2 + & B & \leq 13 \\ & 3E + & C + & 6P_1 + & 6P_2 + & 3B & \leq 45 \\ & 2E + & 4C + & 2P_1 + & 5P_2 + & 3B & \leq 23 \\ & & & P_1 - & P_2 & & = 0 \\ & E, C, P_1, P_2, B \geq 0. \end{array}$$

The optimal solution to the primal and the dual, respectively, together with sensitivity information, is given in Tables 5.4 and 5.5. Use this information to answer the questions that follow.

	Optimal Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
E	0	−3.571	51	3.571	∞
C	2	0	102	16.667	12.5
P_1	0	0	66	37.571	∞
P_2	0	−37.571	66	37.571	∞
B	5	0	89	47	12.5

Table 5.4: The optimal primal solution and its sensitivity with respect to changes in coefficients of the objective function. The last two columns describe the allowed changes in these coefficients for which the same solution remains optimal.

(a) What is the optimal quantity of each service set, and what is the total profit?

(b) Give an economic (not mathematical) interpretation of the optimal dual variables appearing in the sensitivity report, for each of the five constraints.

(c) Should the manufacturer buy an additional 20 lbs. of Clay at \$1.1 per pound?

(d) Suppose that the number of hours available in the dry room decreases by 30. Give a bound for the decrease in the total profit.

(e) In the current model, the number of Primrose produced using method 1 was required to be the same as the number of Primrose produced by method 2. Consider a revision of the model in which this constraint is replaced by the constraint $P_1 - P_2 \geq 0$. In the reformulated problem would the amount of Primrose made by method 1 be positive?

Exercise 5.9 Using the notation of Section 5.2, show that for any positive scalar λ and any $\mathbf{b} \in S$, we have $F(\lambda\mathbf{b}) = \lambda F(\mathbf{b})$. Assume that the dual feasible set is nonempty, so that $F(\mathbf{b})$ is finite.

	Slack Value	Dual Variable	Constr. RHS	Allowable Increase	Allowable Decrease
Clay	130	1.429	130	23.33	43.75
Enamel	9	0	13	∞	4
Dry Rm.	17	0	45	∞	28
Kiln	23	20.143	23	5.60	3.50
Prim.	0	11.429	0	3.50	0

Table 5.5: The optimal dual solution and its sensitivity. The column labeled "slack value" gives us the optimal values of the slack variables associated with each of the primal constraints. The third column simply repeats the right-hand side vector $\mathbf{b}$, while the last two columns describe the allowed changes in the components of $\mathbf{b}$ for which the optimal dual solution remains the same.

Exercise 5.10 Consider the linear programming problem:

$$\begin{array}{ll} \text{minimize} & x_1 + x_2 \\ \text{subject to} & x_1 + 2x_2 = \theta, \\ & x_1, x_2 \geq 0. \end{array}$$

(a) Find (by inspection) an optimal solution, as a function of θ.

(b) Draw a graph showing the optimal cost as a function of θ.

(c) Use the picture in part (b) to obtain the set of all dual optimal solutions, for every value of θ.

Exercise 5.11 Consider the function $g(\theta)$, as defined in the beginning of Section 5.5. Suppose that $g(\theta)$ is linear for $\theta \in [\theta_1, \theta_2]$. Is it true that there exists a unique optimal solution when $\theta_1 < \theta < \theta_2$? Prove or provide a counterexample.

Exercise 5.12 Consider the parametric programming problem discussed in Section 5.5.

(a) Suppose that for some value of θ, there are exactly two distinct basic feasible solutions that are optimal. Show that they must be adjacent.

(b) Let θ^* be a breakpoint of the function $g(\theta)$. Let $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$ be basic feasible solutions, all of which are optimal for $\theta = \theta^*$. Suppose that $\mathbf{x}^1$ is a unique optimal solution for $\theta < \theta^*$, $\mathbf{x}^3$ is a unique optimal solution for $\theta > \theta^*$, and $\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3$ are the only optimal basic feasible solutions for $\theta = \theta^*$. Provide an example to show that $\mathbf{x}^1$ and $\mathbf{x}^3$ need not be adjacent.

Exercise 5.13 Consider the following linear programming problem:

$$\begin{array}{llcl} \text{minimize} & 4x_1 & + 5x_3 & \\ \text{subject to} & 2x_1 + x_2 & - 5x_3 & = 1 \\ & -3x_1 & + 4x_3 + x_4 & = 2 \\ & x_1, x_2, x_3, x_4 \geq 0. & & \end{array}$$

(a) Write down a simplex tableau and find an optimal solution. Is it unique?

(b) Write down the dual problem and find an optimal solution. Is it unique?

(c) Suppose now that we change the vector $\mathbf{b}$ from $\mathbf{b} = (1, 2)$ to $\mathbf{b} = (1 - 2\theta, 2 - 3\theta)$, where θ is a scalar parameter. Find an optimal solution and the value of the optimal cost, as a function of θ. (For all θ, both positive and negative.)

Exercise 5.14 Consider the problem

$$\begin{array}{ll} \text{minimize} & (\mathbf{c} + \theta\mathbf{d})'\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} + \theta\mathbf{f} \\ & \mathbf{x} \geq \mathbf{0}, \end{array}$$

where $\mathbf{A}$ is an $m \times n$ matrix with linearly independent rows. We assume that the problem is feasible and the optimal cost $f(\theta)$ is finite for all values of θ in some interval $[\theta_1, \theta_2]$.

(a) Suppose that a certain basis is optimal for $\theta = -10$ and for $\theta = 10$. Prove that the same basis is optimal for $\theta = 5$.

(b) Show that $f(\theta)$ is a piecewise quadratic function of θ. Give an upper bound on the number of "pieces."

(c) Let $\mathbf{b} = \mathbf{0}$ and $\mathbf{c} = \mathbf{0}$. Suppose that a certain basis is optimal for $\theta = 1$. For what other nonnegative values of θ is that same basis optimal?

(d) Is $f(\theta)$ convex, concave or neither?

Exercise 5.15 Consider the problem

$$\begin{array}{ll} \text{minimize} & \mathbf{c}'\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} + \theta\mathbf{d} \\ & \mathbf{x} \geq \mathbf{0}, \end{array}$$

and let $f(\theta)$ be the optimal cost, as a function of θ.

(a) Let $X(\theta)$ be the set of all optimal solutions, for a given value of θ. For any nonnegative scalar t, define $X(0, t)$ to be the union of the sets $X(\theta)$, $0 \leq \theta \leq t$. Is $X(0, t)$ a convex set? Provide a proof or a counterexample.

(b) Suppose that we remove the nonnegativity constraints $\mathbf{x} \geq \mathbf{0}$ from the problem under consideration. Is $X(0, t)$ a convex set? Provide a proof or a counterexample.

(c) Suppose that $\mathbf{x}^1$ and $\mathbf{x}^2$ belong to $X(0, t)$. Show that there is a continuous path from $\mathbf{x}^1$ to $\mathbf{x}^2$ that is contained within $X(0, t)$. That is, there exists a continuous function $g(\lambda)$ such that $g(\lambda_1) = \mathbf{x}^1$, $g(\lambda_2) = \mathbf{x}^2$, and $g(\lambda) \in X(0, t)$ for all $\lambda \in (\lambda_1, \lambda_2)$.

Exercise 5.16 Consider the parametric programming problem of Section 5.5. Suppose that some basic feasible solution is optimal if and only if θ is equal to some θ^*.

(a) Suppose that the feasible set is unbounded. Is it true that there exist at least three distinct basic feasible solutions that are optimal when $\theta = \theta^*$?

(b) Answer the question in part (a) for the case where the feasible set is bounded.

Exercise 5.17 Consider the parametric programming problem. Suppose that every basic solution encountered by the algorithm is nondegenerate. Prove that the algorithm does not cycle.

5.8 Notes and sources

The material in this chapter, with the exception of Section 5.3, is standard, and can be found in any text on linear programming.

5.1. A more detailed discussion of the results of the production planning case study can be found in Freund and Shannahan (1992).

5.3. The results in this section have beautiful generalizations to the case of nonlinear convex optimization; see, e.g., Rockafellar (1970).

5.5. Anticycling rules for parametric programming can be found in Murty (1983).

Chapter 6

Large scale optimization

Contents

In this chapter, we discuss methods for solving linear programming problems with a large number of variables or constraints. We present the idea of *delayed column generation* whereby we generate a column of the matrix $\mathbf{A}$ only after it has been determined that it can profitably enter the basis. The dual of this idea leads to the *cutting plane*, or *delayed constraint generation* method, in which the feasible set is approximated using only a subset of the constraints, with more constraints added if the resulting solution is infeasible. We illustrate the delayed column generation method by discussing a classical application, the *cutting-stock* problem. Another application is found in *Dantzig-Wolfe decomposition*, which is a method designed for linear programming problems with a special structure. We close with a discussion of stochastic programming, which deals with two-stage optimization problems involving uncertainty. We obtain a large scale linear programming formulation, and we present a decomposition method known as *Benders decomposition*.

6.1 Delayed column generation

Consider the standard form problem

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

with $\mathbf{x} \in \Re^n$ and $\mathbf{b} \in \Re^m$, under the usual assumption that the rows of $\mathbf{A}$ are linearly independent. Suppose that the number of columns is so large that it is impossible to generate and store the entire matrix $\mathbf{A}$ in memory. Experience with large problems indicates that, usually, most of the columns never enter the basis, and we can therefore afford not to ever generate these unused columns. This blends well with the revised simplex method which, at any given iteration, only requires the current basic columns and the column which is to enter the basis. There is only one difficulty that remains to be addressed; namely, we need a method for discovering variables x_i with negative reduced costs $\bar{c}_i$, without having to generate all columns. Sometimes, this can be accomplished by solving the problem

$$\text{minimize} \quad \bar{c}_i, \tag{6.1}$$

where the minimization is over all i. In many instances (e.g., for the formulations to be studied in Sections 6.2 and 6.4), this optimization problem has a special structure: a smallest $\bar{c}_i$ can be found efficiently without computing every $\bar{c}_i$. If the minimum in this optimization problem is greater than or equal to 0, all reduced costs are nonnegative and we have an optimal solution to the original linear programming problem. If on the other hand, the minimum is negative, the variable x_i corresponding to a minimizing index i has negative reduced cost, and the column $\mathbf{A}_i$ can enter the basis.

The key to the above outlined approach is our ability to solve the optimization problem (6.1) efficiently. We will see that in the Dantzig-Wolfe decomposition method, the problem (6.1) is a smaller auxiliary linear programming problem that can be solved using the simplex method. For the cutting-stock problem, the problem (6.1) is a certain discrete optimization problem that can be solved fairly efficiently using special purpose methods. Of course, there are also cases where the problem (6.1) has no special structure and the methodology described here cannot be applied.

A variant involving retained columns

In the delayed column generation method that we have just discussed, the columns that exit the basis are discarded from memory and do not enjoy any special status. In a variant of this method, the algorithm retains in memory all or some of the columns that have been generated in the past, and proceeds in terms of restricted linear programming problems that involve only the retained columns.

We describe the algorithm as a sequence of *master* iterations. At the beginning of a master iteration, we have a basic feasible solution to the original problem, and an associated basis matrix. We search for a variable with negative reduced cost, possibly by minimizing $\bar{c}_i$ over all i; if none is found, the algorithm terminates. Suppose that we have found some j such that $\bar{c}_j < 0$. We then form a collection of columns $\mathbf{A}_i$, $i \in I$, which contains all of the basic columns, the entering column $\mathbf{A}_j$, and possibly some other columns as well. Let us define the *restricted* problem

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in I} c_i x_i \\ \text{subject to} \quad & \sum_{i \in I} \mathbf{A}_i x_i = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{6.2}$$

Recall that the basic variables at the current basic feasible solution to the original problem are among the columns that have been kept in the restricted problem. We therefore have a basic feasible solution to the restricted problem, which can be used as a starting point for its solution. We then perform as many simplex iterations as needed, until the restricted problem is solved to optimality. At that point, we are ready to start with the next master iteration.

The method we have just described is a special case of the revised simplex method, in conjunction with some special rules for choosing the entering variable that give priority to the variables x_i, $i \in I$; it is only when the reduced costs of these variables are all nonnegative (which happens at an optimal solution to the restricted problem) that the algorithm examines the reduced costs of the remaining variables. The motivation is that we may wish to give priority to variables for which the corresponding columns have already been generated and stored in memory, or to variables that are more probable to have negative reduced cost. There are several variants of this method, depending on the manner that the set I is chosen at each iteration.

(a) At one extreme, I is just the set of indices of the current basic variables, together with the entering variable; a variable that exits the basis is immediately dropped from the set I. Since the restricted problem has $m+1$ variables and m constraints, its feasible set is at most one-dimensional, and it gets solved in a single simplex iteration, that is, as soon as the column $\mathbf{A}_j$ enters the basis.

(b) At the other extreme, we let I be the set of indices of all variables that have become basic at some point in the past; equivalently, no variables are ever dropped, and each entering variable is added to I. If the number of master iterations is large, this option can be problematic because the set I keeps growing.

(c) Finally, there are intermediate options in which the set I is kept to a moderate size by dropping from I those variables that have exited the basis in the remote past and have not reentered since.

In the absence of degeneracy, all of the above variants are guaranteed to terminate because they are special cases of the revised simplex method. In the presence of degeneracy, cycling can be avoided by using the revised simplex method in conjunction with the lexicographic tie breaking rule.

6.2 The cutting stock problem

In this section, we discuss the cutting stock problem, which is a classical example of delayed column generation.

Consider a paper company that has a supply of large rolls of paper, of width W. (We assume that W is a positive integer.) However, customer demand is for smaller widths of paper; in particular b_i rolls of width w_i, $i=1,2,\ldots,m$, need to be produced. We assume that $w_i \le W$ for each i, and that each w_i is an integer. Smaller rolls are obtained by slicing a large roll in a certain way, called a *pattern*. For example, a large roll of width 70 can be cut into three rolls of width $w_1=17$ and one roll of width $w_2=15$, with a waste of 4.

In general, a pattern, say the jth pattern, can be represented by a column vector $\mathbf{A}_j$ whose ith entry a_{ij} indicates how many rolls of width w_i are produced by that pattern. For example, the pattern described earlier is represented by the vector $(3,1,0,\ldots,0)$. For a vector $(a_{1j},\ldots,a_{mj})$ to be a representation of a feasible pattern, its components must be nonnegative integers and we must also have

$$\sum_{i=1}^{m} a_{ij} w_i \le W. \tag{6.3}$$

Let n be the number of all feasible patterns and consider the $m \times n$ matrix $\mathbf{A}$ with columns $\mathbf{A}_j$, $j=1,\ldots,n$. Note that n can be a very large number.

The goal of the company is to minimize the number of large rolls used while satisfying customer demand. Let x_j be the number of large rolls cut according to pattern j. Then, the problem under consideration is

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{n} x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j = b_i, \qquad i=1,\ldots,m, \\ & x_j \ge 0, \qquad j=1,\ldots,n. \end{aligned} \tag{6.4}$$

Naturally, each x_j should be an integer and we have an integer programming problem. However, an optimal solution to the linear programming problem (6.4) often provides a feasible solution to the integer programming problem (by rounding or other ad hoc methods), which is fairly close to optimal, at least if the demands b_i are reasonably large (cf. Exercise 6.1).

Solving the linear programming problem (6.4) is a difficult computational task: even if m is comparatively small, the number of feasible patterns n can be huge, so that forming the coefficient matrix $\mathbf{A}$ in full is impractical. However, we will now show that the problem can be solved efficiently, by using the revised simplex method and by generating columns of $\mathbf{A}$ as needed rather than in advance.

Finding an initial basic feasible solution is easy for this problem. For $j=1,\ldots,m$, we may let the jth pattern consist of one roll of width w_j and none of the other widths. Then, the first m columns of $\mathbf{A}$ form a basis that leads to a basic feasible solution. (In fact, the corresponding basis matrix is the identity.)

Suppose now that we have a basis matrix $\mathbf{B}$ and an associated basic feasible solution, and that we wish to carry out the next iteration of the revised simplex method. Because the cost coefficient of every variable x_j is unity, every component of the vector $\mathbf{c}_B$ is equal to 1. We compute the simplex multipliers $\mathbf{p}' = \mathbf{c}'_B \mathbf{B}^{-1}$. Next, instead of computing the reduced cost $\bar{c}_j = 1 - \mathbf{p}'\mathbf{A}_j$ associated with every column (pattern) $\mathbf{A}_j$, we consider the problem of minimizing $(1-\mathbf{p}'\mathbf{A}_j)$ over all j. This is the same as maximizing $\mathbf{p}'\mathbf{A}_j$ over all j. If the maximum is less than or equal to 1, all reduced costs are nonnegative and we have an optimal solution. If on the other hand, the maximum is greater than 1, the column $\mathbf{A}_j$ corresponding to a maximizing j has negative reduced cost and enters the basis.

We are now left with the task of finding a pattern j that maximizes $\mathbf{p}'\mathbf{A}_j$. Given our earlier description of what constitutes an admissible pattern [cf. Eq. (6.3)], we are faced with the problem

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{m} p_i a_i \\ \text{subject to} \quad & \sum_{i=1}^{m} w_i a_i \le W \\ & a_i \ge 0, \qquad i = 1, \ldots, m, \\ & a_i \text{ integer}, \qquad i = 1, \ldots, m. \end{aligned} \tag{6.5}$$

This problem is called the *integer knapsack* problem. (Think of p_i as the value, and w_i as the weight of the ith item; we seek to fill a knapsack and maximize its value without the total weight exceeding W). Solving the knapsack problem requires some effort, but for the range of numbers that arise in the cutting stock problem, this can be done fairly efficiently.

One possible algorithm for solving the knapsack problem, based on *dynamic programming*, is as follows. Let $F(v)$ denote the optimal objective value in the problem (6.5), when W is replaced by v, and let us use the convention $F(v) = 0$ when $v < 0$. Let $w_{\min} = \min_i w_i$. If $v < w_{\min}$, then clearly $F(v) = 0$. For $v \ge w_{\min}$, we have the recursion

$$F(v) = \max_{i=1,\ldots,m} \{F(v - w_i) + p_i\}. \tag{6.6}$$

For an interpretation of this recursion, note that a knapsack of weight at most v is obtained by first filling the knapsack with weight at most $v - w_i$, and then adding an item of weight w_i. The knapsack of weight at most $v - w_i$ should be filled so that we obtain the maximum value, which is $F(v - w_i)$, and the ith item should be chosen so that the total value $F(v - w_i) + p_i$ is maximized. Using the recursion (6.6), $F(v)$ can be computed for $v = w_{\min}, w_{\min} + 1, \ldots, W$. In addition, an optimal solution is obtained by backtracking if a record of the maximizing index i is kept at each step. The computational complexity of this procedure is $O(mW)$ because the recursion (6.6) is to be carried out for $O(W)$ different values of v, each time requiring $O(m)$ arithmetic operations.

The dynamic programming methodology is discussed in more generality in Section 11.3, where it is also applied to a somewhat different variant of the knapsack problem. The knapsack problem can also be solved using the branch and bound methodology, developed in Section 11.2.

6.3 Cutting plane methods

Delayed column generation methods, when viewed in terms of the dual variables, can be described as *delayed constraint generation*, or *cutting plane* methods. In this section, we develop this alternative perspective.

Consider the problem

$$\begin{aligned} \text{maximize} \quad & \mathbf{p}'\mathbf{b} \\ \text{subject to} \quad & \mathbf{p}'\mathbf{A}_i \le c_i, \qquad i = 1, \ldots, n, \end{aligned} \tag{6.7}$$

which is the dual of the standard form problem considered in Section 6.1. Once more, we assume that it is impossible to generate and store each one of the vectors $\mathbf{A}_i$, because the number n is very large. Instead of dealing with all n of the dual constraints, we consider a subset I of $\{1, \ldots, n\}$, and form the *relaxed dual* problem

$$\begin{aligned} \text{maximize} \quad & \mathbf{p}'\mathbf{b} \\ \text{subject to} \quad & \mathbf{p}'\mathbf{A}_i \le c_i, \qquad i \in I, \end{aligned} \tag{6.8}$$

which we solve to optimality. Let $\mathbf{p}^*$ be an optimal basic feasible solution to the relaxed dual problem. There are two possibilities.

(a) Suppose that $\mathbf{p}^*$ is a feasible solution to the original problem (6.7). Any other feasible solution $\mathbf{p}$ to the original problem (6.7) is also feasible for the relaxed problem (6.8), because the latter has fewer constraints. Therefore, by the optimality of $\mathbf{p}^*$ for the problem (6.8), we have $\mathbf{p}'\mathbf{b} \le (\mathbf{p}^*)'\mathbf{b}$. Therefore, $\mathbf{p}^*$ is an optimal solution to the original problem (6.7), and we can terminate the algorithm.

(b) If $\mathbf{p}^*$ is infeasible for the problem (6.7), we find a violated constraint, add it to the constraints of the relaxed dual problem, and continue similarly. See Figure 6.1 for an illustration.

In order to carry out this algorithm, we need a method for checking whether a vector $\mathbf{p}^*$ is a feasible solution to the original dual problem (6.7). Second, if $\mathbf{p}^*$ is dual infeasible, we need an efficient method for identifying a violated constraint. (This is known as the *separation problem*, because it amounts to finding a hyperplane that separates $\mathbf{p}^*$ from the dual feasible set, and is discussed further in Section 8.5.) One possibility is to formulate and solve the optimization problem

$$\text{minimize} \quad c_i - (\mathbf{p}^*)'\mathbf{A}_i \tag{6.9}$$

over all i. If the optimal value in this problem is nonnegative, we have a feasible (and, therefore, optimal) solution to the original dual problem; if it is negative, then an optimizing i satisfies $c_i < (\mathbf{p}^*)'\mathbf{A}_i$, and we have identified a violated constraint. The success of this approach hinges on our ability to solve the problem (6.9) efficiently; fortunately, this is sometimes possible. In addition, there are cases where the optimization problem (6.9) is not easily solved but one can test for feasibility and identify violated constraints using other means. (See, e.g., Section 11.1 for applications of the cutting plane method to integer programming problems.)

It should be apparent at this point that applying the cutting plane method to the dual problem is identical to applying the delayed column

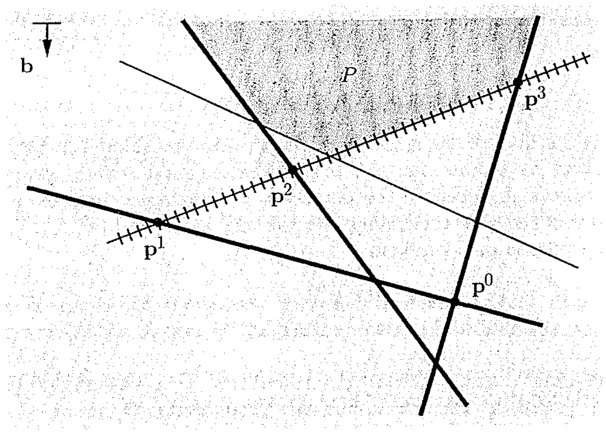

Figure 6.1: A polyhedron P defined in terms of several inequality constraints. Let the vector $\mathbf{b}$ point downwards, so that maximizing $\mathbf{p}'\mathbf{b}$ is the same as looking for the lowest point. We start with the constraints indicated by the thicker lines, and the optimal solution to the relaxed dual problem is $\mathbf{p}^0$. The vector $\mathbf{p}^0$ is infeasible and we identify the constraint indicated by a thatched line as a violated one. We incorporate this constraint in the relaxed dual problem, and solve the new relaxed dual problem to optimality, to arrive at the vector $\mathbf{p}^2$.

generation method to the primal. For example, minimizing $c_i - (\mathbf{p}^*)'\mathbf{A}_i$ in order to find a violated dual constraint is identical to minimizing $\bar{c}_i$ in order to find a primal variable with negative reduced cost. Furthermore, the relaxed dual problem (6.8) is simply the dual of the restricted primal problem (6.2) formed in Section 6.1.

The cutting plane method, as described here, corresponds to the variant of delayed column generation in which all columns generated by the algorithm are retained, and the set I grows with each iteration. As discussed in Section 6.1, a possible alternative is to drop some of the elements of I; for example, we could drop those constraints that have not been active for some time.

If we take the idea of dropping old dual constraints and carry it to the extreme, we obtain a variant of the cutting plane method whereby, at each stage, we add one violated constraint, move to a new $\mathbf{p}$ vector, and remove a constraint that has been rendered inactive at the new vector.

Example 6.1 Consider Figure 6.1 once more. We start at $\mathbf{p}^0$ and let I consist of the two constraints that are active at $\mathbf{p}^0$. The constraint corresponding to

the thatched line is violated and we add it to the set I. At this point, the set I consists of three dual constraints, and the relaxed dual problem has exactly three basic solutions, namely the points $\mathbf{p}^0$, $\mathbf{p}^1$, and $\mathbf{p}^3$. We maximize $\mathbf{p}'\mathbf{b}$ subject to these three constraints, and the vector $\mathbf{p}^1$ is chosen. At this point, the constraint that goes through $\mathbf{p}^0$ and $\mathbf{p}^3$ is satisfied, but has been rendered inactive. We drop this constraint from I, which leaves us with the two constraints through the point $\mathbf{p}^1$. Since $\mathbf{p}^1$ is infeasible, we can now identify another violated constraint and continue similarly.

Since the cutting plane method is simply the delayed column generation method, viewed from a different angle, there is no need to provide implementation details. While the algorithm is easily visualized in terms of cutting planes and the dual problem, the computations can be carried out using the revised simplex method on the primal problem, in the standard fashion.

We close by noting that in some occasions, we may be faced with a primal problem (not in standard form) that has relatively few variables but a very large number of constraints. In that case, it makes sense to apply the cutting plane algorithm to the primal; equivalently, we can form the dual problem and solve it using delayed column generation.

6.4 Dantzig-Wolfe decomposition

Consider a linear programming problem of the form

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}_1'\mathbf{x}_1 + \mathbf{c}_2'\mathbf{x}_2 \\ \text{subject to} \quad & \mathbf{D}_1\mathbf{x}_1 + \mathbf{D}_2\mathbf{x}_2 = \mathbf{b}_0 \\ & \mathbf{F}_1\mathbf{x}_1 = \mathbf{b}_1 \\ & \mathbf{F}_2\mathbf{x}_2 = \mathbf{b}_2 \\ & \mathbf{x}_1, \mathbf{x}_2 \geq \mathbf{0}. \end{aligned} \tag{6.10}$$

Suppose that $\mathbf{x}_1$ and $\mathbf{x}_2$ are vectors of dimensions n_1 and n_2, respectively, and that $\mathbf{b}_0$, $\mathbf{b}_1$, $\mathbf{b}_2$ have dimensions m_0, m_1, m_2, respectively. Thus, besides nonnegativity constraints, $\mathbf{x}_1$ satisfies m_1 constraints, $\mathbf{x}_2$ satisfies m_2 constraints, and $\mathbf{x}_1, \mathbf{x}_2$ together satisfy m_0 coupling constraints. Here, $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{F}_1$, $\mathbf{F}_2$ are matrices of appropriate dimensions.

Problems with the structure we have just described arise in several applications. For example, $\mathbf{x}_1$ and $\mathbf{x}_2$ could be decision variables associated with two divisions of the same firm. There are constraints tied to each division, and there are also some coupling constraints representing shared resources, such as a total budget. Often, the number of coupling constraints is a small fraction of the total. We will now proceed to develop a decomposition method tailored to problems of this type.

Reformulation of the problem

Our first step is to introduce an equivalent problem, with fewer equality constraints, but many more variables.

For $i = 1, 2$, we define

$$P_i = \{\mathbf{x}_i \geq \mathbf{0} \mid \mathbf{F}_i\mathbf{x}_i = \mathbf{b}_i\},$$

and we assume that P_1 and P_2 are nonempty. Then, the problem can be rewritten as

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}_1'\mathbf{x}_1 + \mathbf{c}_2'\mathbf{x}_2 \\ \text{subject to} \quad & \mathbf{D}_1\mathbf{x}_1 + \mathbf{D}_2\mathbf{x}_2 = \mathbf{b}_0 \\ & \mathbf{x}_1 \in P_1 \\ & \mathbf{x}_2 \in P_2. \end{aligned}$$

For $i = 1, 2$, let $\mathbf{x}_i^j$, $j \in J_i$, be the extreme points of P_i. Let also $\mathbf{w}_i^k$, $k \in K_i$, be a complete set of extreme rays of P_i. Using the resolution theorem (Theorem 4.15 in Section 4.9), any element $\mathbf{x}_i$ of P_i can be represented in the form

$$\mathbf{x}_i = \sum_{j \in J_i} \lambda_i^j \mathbf{x}_i^j + \sum_{k \in K_i} \theta_i^k \mathbf{w}_i^k,$$

where the coefficients λ_i^j and θ_i^k are nonnegative and satisfy

$$\sum_{j \in J_i} \lambda_i^j = 1, \qquad i = 1, 2.$$

The original problem (6.10) can be now reformulated as

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in J_1} \lambda_1^j \mathbf{c}_1'\mathbf{x}_1^j + \sum_{k \in K_1} \theta_1^k \mathbf{c}_1'\mathbf{w}_1^k + \sum_{j \in J_2} \lambda_2^j \mathbf{c}_2'\mathbf{x}_2^j + \sum_{k \in K_2} \theta_2^k \mathbf{c}_2'\mathbf{w}_2^k \\ \text{subject to} \quad & \sum_{j \in J_1} \lambda_1^j \mathbf{D}_1\mathbf{x}_1^j + \sum_{k \in K_1} \theta_1^k \mathbf{D}_1\mathbf{w}_1^k + \sum_{j \in J_2} \lambda_2^j \mathbf{D}_2\mathbf{x}_2^j \\ & \qquad + \sum_{k \in K_2} \theta_2^k \mathbf{D}_2\mathbf{w}_2^k = \mathbf{b}_0 \qquad (6.11) \\ & \sum_{j \in J_1} \lambda_1^j = 1 \qquad (6.12) \\ & \sum_{j \in J_2} \lambda_2^j = 1 \qquad (6.13) \\ & \lambda_i^j \geq 0, \ \theta_i^k \geq 0, \qquad \forall\, i, j, k. \end{aligned}$$

This problem will be called the *master* problem. It is equivalent to the original problem (6.10) and is a linear programming problem in standard form, with decision variables λ_i^j and θ_i^k. An alternative notation for the

equality constraints (6.11)-(6.13) that shows more clearly the structure of the column associated with each variable is

$$\sum_{j \in J_1} \lambda_1^j \begin{bmatrix} \mathbf{D}_1\mathbf{x}_1^j \\ 1 \\ 0 \end{bmatrix} + \sum_{j \in J_2} \lambda_2^j \begin{bmatrix} \mathbf{D}_2\mathbf{x}_2^j \\ 0 \\ 1 \end{bmatrix} + \sum_{k \in K_1} \theta_1^k \begin{bmatrix} \mathbf{D}_1\mathbf{w}_1^k \\ 0 \\ 0 \end{bmatrix} + \sum_{k \in K_2} \theta_2^k \begin{bmatrix} \mathbf{D}_2\mathbf{w}_2^k \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_0 \\ 1 \\ 1 \end{bmatrix}.$$

The decomposition algorithm

In contrast to the original problem, which had $m_0 + m_1 + m_2$ equality constraints, the master problem has only $m_0 + 2$ equality constraints. On the other hand, the number of decision variables in the master problem could be astronomical, because the number of extreme points and rays is usually exponential in the number of variables and constraints. Because of the enormous number of variables in the master problem, we need to use the revised simplex method which, at any given iteration, involves only m_0+2 basic variables and a basis matrix of dimensions $(m_0+2) \times (m_0+2)$.

Suppose that we have a basic feasible solution to the master problem, associated with a basis matrix $\mathbf{B}$. We assume that the inverse basis matrix $\mathbf{B}^{-1}$ is available, as well as the dual vector $\mathbf{p}' = \mathbf{c}_B'\mathbf{B}^{-1}$. Since we have m_0+2 equality constraints, the vector $\mathbf{p}$ has dimension m_0+2. The first m_0 components of $\mathbf{p}$, to be denoted by $\mathbf{q}$, are the dual variables associated with the constraints (6.11). The last two components, to be denoted by r_1 and r_2, are the dual variables associated with the "convexity" constraints (6.12) and (6.13), respectively. In particular, $\mathbf{p} = (\mathbf{q}, r_1, r_2)$.

In order to decide whether the current basic feasible solution is optimal, we need to examine the reduced costs of the different variables and check whether any one of them is negative. The cost coefficient of a variable λ_1^j is $\mathbf{c}_1'\mathbf{x}_1^j$. Therefore, the reduced cost of the variable λ_1^j is given by

$$\mathbf{c}_1'\mathbf{x}_1^j - \begin{bmatrix}\mathbf{q}' & r_1 & r_2\end{bmatrix} \begin{bmatrix} \mathbf{D}_1\mathbf{x}_1^j \\ 1 \\ 0 \end{bmatrix} = (\mathbf{c}_1' - \mathbf{q}'\mathbf{D}_1)\mathbf{x}_1^j - r_1.$$

Similarly, the cost coefficient of the variable θ_1^k is $\mathbf{c}_1'\mathbf{w}_1^k$. Therefore, its reduced cost is

$$\mathbf{c}_1'\mathbf{w}_1^k - \begin{bmatrix}\mathbf{q}' & r_1 & r_2\end{bmatrix} \begin{bmatrix} \mathbf{D}_1\mathbf{w}_1^k \\ 0 \\ 0 \end{bmatrix} = (\mathbf{c}_1' - \mathbf{q}'\mathbf{D}_1)\mathbf{w}_1^k.$$

We now introduce the most critical idea in the decomposition algorithm. Instead of evaluating the reduced cost of every variable λ_1^j and θ_1^k, and checking its sign, we form the linear programming problem

$$\begin{array}{ll} \text{minimize} & (\mathbf{c}_1' - \mathbf{q}'\mathbf{D}_1)\mathbf{x}_1 \\ \text{subject to} & \mathbf{x}_1 \in P_1, \end{array}$$

called the first *subproblem*, which we solve by means of the simplex method.

There are three possibilities to consider.

(a) If the optimal cost in the subproblem is $-\infty$, then, upon termination, the simplex method provides us with an extreme ray $\mathbf{w}_1^k$ that satisfies $(\mathbf{c}_1' - \mathbf{q}'\mathbf{D}_1)\mathbf{w}_1^k < 0$ (see the discussion at the end of Section 4.8). In this case, the reduced cost of the variable θ_1^k is negative. At this point, we can generate the column

$$\begin{bmatrix} \mathbf{D}_1\mathbf{w}_1^k \\ 0 \\ 0 \end{bmatrix}$$

associated with θ_1^k, and have it enter the basis in the master problem.

(b) If the optimal cost in the subproblem is finite and smaller than r_1, then, upon termination, the simplex method provides us with an extreme point $\mathbf{x}_1^j$ that satisfies $(\mathbf{c}_1' - \mathbf{q}'\mathbf{D}_1)\mathbf{x}_1^j < r_1$. In this case, the reduced cost of the variable λ_1^j is negative. At this point, we can generate the column

$$\begin{bmatrix} \mathbf{D}_1\mathbf{x}_1^j \\ 1 \\ 0 \end{bmatrix}$$

associated with λ_1^j, and have it enter the basis in the master problem.

(c) Finally, if the optimal cost in the subproblem is finite and no smaller than r_1, this implies that $(\mathbf{c}_1' - \mathbf{q}'\mathbf{D}_1)\mathbf{x}_1^j \geq r_1$ for all extreme points $\mathbf{x}_1^j$, and $(\mathbf{c}_1' - \mathbf{q}'\mathbf{D}_1)\mathbf{w}_1^k \geq 0$ for all extreme rays $\mathbf{w}_1^k$. In this case, the reduced cost of every variable λ_1^j or θ_1^k is nonnegative.

The same approach is followed for checking the reduced costs of the variables λ_2^j and θ_2^k: we form the second subproblem

$$\begin{array}{ll} \text{minimize} & (\mathbf{c}_2' - \mathbf{q}'\mathbf{D}_2)\mathbf{x}_2 \\ \text{subject to} & \mathbf{x}_2 \in P_2, \end{array}$$

and solve it using the simplex method. Either the optimal cost is greater than or equal to r_2 and all reduced costs are nonnegative, or we find a variable λ_2^j or θ_2^k whose reduced cost is negative and can enter the basis.

The resulting algorithm is summarized below.

Dantzig-Wolfe decomposition algorithm

1. A typical iteration starts with a total of m_0+2 extreme points and extreme rays of P_1, P_2, which lead to a basic feasible solution to the master problem, the corresponding inverse basis matrix $\mathbf{B}^{-1}$, and the dual vector $\mathbf{p}' = (\mathbf{q}, r_1, r_2)' = \mathbf{c}_B'\mathbf{B}^{-1}$.
2. Form and solve the two subproblems. If the optimal cost in the first subproblem is no smaller than r_1 and the optimal cost in the second subproblem is no smaller than r_2, then all reduced costs in the master problem are nonnegative, we have an optimal solution, and the algorithm terminates.
3. If the optimal cost in the ith subproblem is $-\infty$, we obtain an extreme ray $\mathbf{w}_i^k$, associated with a variable θ_i^k whose reduced cost is negative; this variable can enter the basis in the master problem.
4. If the optimal cost in the ith subproblem is finite and less than r_i, we obtain an extreme point $\mathbf{x}_i^j$, associated with a variable λ_i^j whose reduced cost is negative; this variable can enter the basis in the master problem.
5. Having chosen a variable to enter the basis, generate the column associated with that variable, carry out an iteration of the revised simplex method for the master problem, and update $\mathbf{B}^{-1}$ and $\mathbf{p}$.

We recognize delayed column generation as the centerpiece of the decomposition algorithm. Even though the master problem can have a huge number of columns, a column is generated only after it is found to have negative reduced cost and is about to enter the basis. Note that the subproblems are smaller linear programming problems that are employed as an economical search method for discovering columns with negative reduced costs.

As discussed in Section 6.1, we can also use a variant whereby all columns that have been generated in the past are retained. In this case, Step 5 of the algorithm has to be modified. Instead of carrying out a single simplex iteration, we solve a restricted master problem to optimality. This restricted problem has the same structure as the master problem, except that it only involves the columns that have been generated so far.

Economic interpretation

We provide an appealing economic interpretation of the Dantzig-Wolfe decomposition method. We have an organization with two divisions that have to meet a common objective, reflected in the coupling constraint $\mathbf{D}_1\mathbf{x}_1 + \mathbf{D}_2\mathbf{x}_2 = \mathbf{b}_0$. A central planner assigns a value of $\mathbf{q}$ for each unit of contribution towards the common objective. Division i is interested in minimizing $\mathbf{c}_i'\mathbf{x}_i$ subject to its own constraints. However, any choice of $\mathbf{x}_i$ contributes $\mathbf{D}_i\mathbf{x}_i$ towards the common objective and has therefore a value of $\mathbf{q}'\mathbf{D}_i\mathbf{x}_i$. This leads division i to minimize $(\mathbf{c}_i' - \mathbf{q}'\mathbf{D}_i)\mathbf{x}_i$ (cost minus value) subject to its local constraints. The optimal solution to the division's subproblem can be viewed as a proposal to the central planner. The central planner uses these proposals and combines them (optimally) with preexisting proposals to form a feasible solution for the overall problem. Based on this feasible solution (which is a basic feasible solution to the master problem), the values $\mathbf{q}$ are reassessed and the process is repeated.

Applicability of the method

It should be clear that there is nothing special about having exactly two subproblems in the Dantzig-Wolfe decomposition method. In particular, the method generalizes in a straightforward manner to problems of the form

$$\begin{array}{lll} \text{minimize} & \mathbf{c}_1'\mathbf{x}_1 + \mathbf{c}_2'\mathbf{x}_2 + \cdots + \mathbf{c}_t'\mathbf{x}_t & \\ \text{subject to} & \mathbf{D}_1\mathbf{x}_1 + \mathbf{D}_2\mathbf{x}_2 + \cdots + \mathbf{D}_t\mathbf{x}_t = \mathbf{b}_0 & \\ & \mathbf{F}_i\mathbf{x}_i = \mathbf{b}_i, & i = 1, 2, \ldots, t, \\ & \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t \geq \mathbf{0}. & \end{array}$$

The only difference is that at each iteration of the revised simplex method for the master problem, we may have to solve t subproblems.

In fact, the method is applicable even if $t = 1$, as we now discuss. Consider the linear programming problem

$$\begin{array}{ll} \text{minimize} & \mathbf{c}'\mathbf{x} \\ \text{subject to} & \mathbf{D}\mathbf{x} = \mathbf{b}_0 \\ & \mathbf{F}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{array}$$

in which the equality constraints have been partitioned into two sets, and define the polyhedron $P = \{\mathbf{x} \geq \mathbf{0} \mid \mathbf{F}\mathbf{x} = \mathbf{b}\}$. By expressing each element of P in terms of extreme points and extreme rays, we obtain a master problem with a large number of columns, but a smaller number of equality constraints. Searching for columns with negative reduced cost in the master problem is then accomplished by solving a single subproblem, which is a minimization over the set P. This approach can be useful if the subproblem has a special structure and can be solved very fast.

Throughout our development, we have been assuming that all constraints are in standard form and, in particular, the feasible sets P_i of the subproblems are also in standard form. This is hardly necessary. For example, if we assume that the sets P_i have at least one extreme point, the resolution theorem and the same line of development applies.

Examples

We now consider some examples and go through the details of the algorithm. In order to avoid excessive bookkeeping, our first example involves a single subproblem.

Example 6.2 Consider the problem

$$\begin{array}{ll} \text{minimize} & -4x_1 - x_2 - 6x_3 \\ \text{subject to} & 3x_1 + 2x_2 + 4x_3 = 17 \\ & 1 \leq x_1 \leq 2 \\ & 1 \leq x_2 \leq 2 \\ & 1 \leq x_3 \leq 2. \end{array}$$

We divide the constraints into two groups: the first group consists of the constraint $\mathbf{D}\mathbf{x} = \mathbf{b}_0$, where $\mathbf{D}$ is the 1×3 matrix $\mathbf{D} = [3\ \ 2\ \ 4]$, and where $\mathbf{b}_0 = 17$; the second group is the constraint $\mathbf{x} \in P$, where $P = \{\mathbf{x} \in \Re^3 \mid 1 \leq x_i \leq 2,\ i = 1, 2, 3\}$. Note that P has eight extreme points; furthermore, it is bounded and, therefore, has no extreme rays. The master problem has two equality constraints, namely

$$\sum_{j=1}^{8} \lambda^j \mathbf{D}\mathbf{x}^j = 17,$$
$$\sum_{j=1}^{8} \lambda^j = 1;$$

where $\mathbf{x}^j$ are the extreme points of P. The columns of the constraint matrix in the master problem are of the form $(\mathbf{D}\mathbf{x}^j, 1)$. Let us pick two of the extreme points of P, say, $\mathbf{x}^1 = (2, 2, 2)$ and $\mathbf{x}^2 = (1, 1, 2)$, and let the corresponding variables λ^1 and λ^2 be our initial basic variables. We have $\mathbf{D}\mathbf{x}^1 = 18$, $\mathbf{D}\mathbf{x}^2 = 13$, and, therefore, the corresponding basis matrix is

$$\mathbf{B} = \begin{bmatrix} 18 & 13 \\ 1 & 1 \end{bmatrix};$$

its inverse is

$$\mathbf{B}^{-1} = \begin{bmatrix} 0.2 & -2.6 \\ -0.2 & 3.6 \end{bmatrix}.$$

We form the product of $\mathbf{B}^{-1}$ with the vector $(17, 1)$. The result, which is $(0.8, 0.2)$, gives us the values of the basic variables λ^1, λ^2. Since these values are nonnegative, we have a basic feasible solution to the master problem.

We now determine the simplex multipliers. Recalling that the cost of λ^j is $\mathbf{c}'\mathbf{x}^j$, we have

$$c_{B(1)} = \mathbf{c}'\mathbf{x}^1 = \begin{bmatrix} -4 & -1 & -6 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} = -22,$$

and

$$c_{B(2)} = \mathbf{c}'\mathbf{x}^2 = \begin{bmatrix} -4 & -1 & -6 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = -17.$$

We therefore have

$$\mathbf{p}' = \begin{bmatrix} \mathbf{q}' & r \end{bmatrix} = \mathbf{c}_B'\mathbf{B}^{-1} = \begin{bmatrix} -22 & -17 \end{bmatrix}\mathbf{B}^{-1} = \begin{bmatrix} -1 & -4 \end{bmatrix}.$$

We now form the subproblem. We are to minimize $(\mathbf{c}' - \mathbf{q}'\mathbf{D})\mathbf{x}$ subject to $\mathbf{x} \in P$. We have

$$\mathbf{c}' - \mathbf{q}'\mathbf{D} = \begin{bmatrix} -4 & -1 & -6 \end{bmatrix} - (-1)\begin{bmatrix} 3 & 2 & 4 \end{bmatrix} = [-1 \;\; 1 \;\; -2],$$

and the optimal solution is $\mathbf{x} = (2, 1, 2)$. This is a new extreme point of P, which we will denote by $\mathbf{x}^3$. The optimal cost in the subproblem is -5, and is less than r, which is -4. It follows that the reduced cost of the variable λ^3 is negative, and this variable can enter the basis. At this point, we generate the column corresponding to λ^3. Since $\mathbf{D}\mathbf{x}^3 = 16$, the corresponding column, call it $\mathbf{g}$, is $(16, 1)$. We form the vector $\mathbf{u} = \mathbf{B}^{-1}\mathbf{g}$, which is found to be $(0.6, 0.4)$. In order to determine which variable exits the basis, we form the ratios $\lambda^1/u_1 = 0.8/0.6$ and $\lambda^2/u_2 = 0.2/0.4$. The second ratio is smaller and λ^2 exits the basis. We now have a new basis

$$\mathbf{B} = \begin{bmatrix} 18 & 16 \\ 1 & 1 \end{bmatrix};$$

its inverse is

$$\mathbf{B}^{-1} = \begin{bmatrix} 0.5 & -8 \\ -0.5 & 9 \end{bmatrix}.$$

We form the product of $\mathbf{B}^{-1}$ with the vector $(17, 1)$ and determine the values of the basic variables, which are $\lambda^1 = 0.5$ and $\lambda^3 = 0.5$. The new value of $c_{B(2)}$ is $\mathbf{c}'\mathbf{x}^3 = -21$. Once more, we compute $[\mathbf{q}' \;\; r] = \mathbf{c}_B'\mathbf{B}^{-1}$, which is $(-0.5, -13)'$.

We now go back to the subproblem. We have

$$\mathbf{c}' - \mathbf{q}'\mathbf{D} = \begin{bmatrix} -4 & -1 & -6 \end{bmatrix} - (-0.5)\begin{bmatrix} 3 & 2 & 4 \end{bmatrix} = \begin{bmatrix} -2.5 & 0 & -4 \end{bmatrix}.$$

We minimize $(\mathbf{c}' - \mathbf{q}'\mathbf{D})\mathbf{x}$ over P. We find that $(2, 2, 2)$ is an optimal solution, and the optimal cost is equal to -13. Since this is the same as the value of r, we conclude that the reduced cost of every λ^i is nonnegative, and we have an optimal solution to the master problem.

In terms of the variables x_i, the optimal solution is

$$\mathbf{x} = \frac{1}{2}\mathbf{x}^1 + \frac{1}{2}\mathbf{x}^3 = \begin{bmatrix} 2 \\ 1.5 \\ 2 \end{bmatrix}.$$

The progress of the algorithm is illustrated in Figure 6.2.

As shown in Figure 6.2, even though the optimal solution is an extreme point of the feasible set in $\mathbf{x}$-space, feasible solutions generated in the course of the algorithm (e.g., the point A) are not extreme points. Another illustration that conveys the same message is provided by Figure 6.3. Notice the similarity with our discussion of the column geometry in Section 3.6.

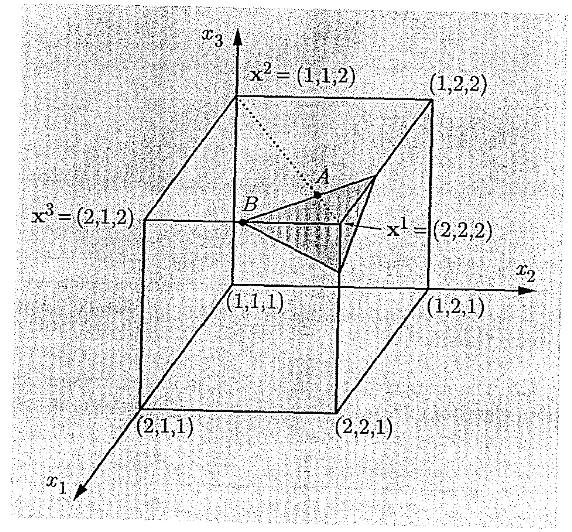

Figure 6.2: Illustration of Example 6.2, in terms of the variables x_i in the original problem. The cube shown is the set P. The feasible set is the intersection of the cube with the hyperplane $3x_1 + 2x_2 + 4x_3 = 17$, and corresponds to the shaded triangle. Under the first basis considered, we have a feasible solution which is a convex combination of the extreme points $\mathbf{x}^1$, $\mathbf{x}^2$, namely, point A. At the next step, the extreme point $\mathbf{x}^3$ is introduced. If λ^1 were to become nonbasic, we would be dealing with convex combinations of $\mathbf{x}^2$ and $\mathbf{x}^3$, and we would not be able to satisfy the constraint $3x_1 + 2x_2 + 4x_3 = 17$. This provides a geometric explanation of why λ^1 must stay and λ^2 must exit the basis. The new basic feasible solution corresponds to the point B, is a convex combination of $\mathbf{x}^1$ and $\mathbf{x}^3$, and was found to be optimal.

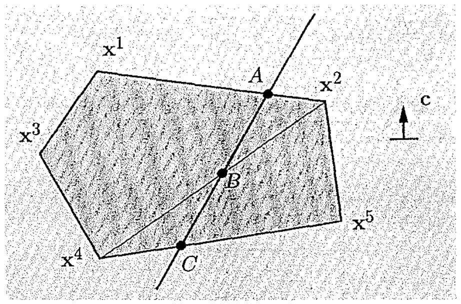

Figure 6.3: Another illustration of the geometry of Dantzig-Wolfe decomposition. Consider the case where there is a single subproblem whose feasible set has extreme points $\mathbf{x}^1, \ldots, \mathbf{x}^5$, and a single coupling equality constraint which corresponds to the line shown in the figure. The algorithm is initialized at point A and follows the path A, B, C, with point C being an optimal solution.

Example 6.3 The purpose of this example is to illustrate the behavior of the decomposition algorithm when the feasible set of a subproblem is unbounded.

Consider the linear programming problem

$$\begin{aligned} \text{minimize} \quad & -5x_1 + x_2 \\ \text{subject to} \quad & x_1 \leq 8 \\ & x_1 - x_2 \leq 4 \\ & 2x_1 - x_2 \leq 10 \\ & x_1, x_2 \geq 0. \end{aligned}$$

The feasible set is shown in Figure 6.4.

We associate a slack variable x_3 with the first constraint and obtain the problem

$$\begin{aligned} \text{minimize} \quad & -5x_1 + x_2 \\ \text{subject to} \quad & x_1 + x_3 = 8 \\ & x_1 - x_2 \leq 4 \\ & 2x_1 - x_2 \leq 10 \\ & x_1, x_2 \geq 0 \\ & x_3 \geq 0. \end{aligned}$$

We view the constraint $x_1 + x_3 = 8$ as a coupling constraint and let

$$P_1 = \Big\{(x_1, x_2) \;\Big|\; x_1 - x_2 \leq 4,\ 2x_1 - x_2 \leq 10,\ x_1, x_2 \geq 0\Big\},$$

$$P_2 = \{x_3 \mid x_3 \geq 0\}.$$

We therefore have two subproblems, although the second subproblem has a very simple feasible set.

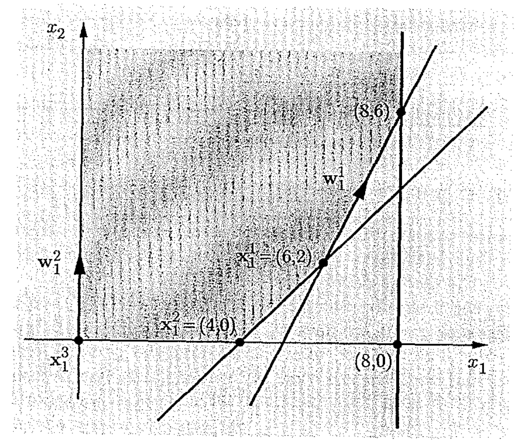

Figure 6.4: Illustration of Example 6.3. The algorithm starts at $(x_1, x_2) = (6, 2)$ and after one master iteration reaches point $(x_1, x_2) = (8, 6)$, which is an optimal solution.

The set P_1 is the same as the set shown in Figure 6.4, except that the constraint $x_1 \leq 8$ is absent. Thus, P_1 has three extreme points, namely, $\mathbf{x}_1^1 = (6, 2)$, $\mathbf{x}_1^2 = (4, 0)$, $\mathbf{x}_1^3 = (0, 0)$, and two extreme rays, namely, $\mathbf{w}_1^1 = (1, 2)$ and $\mathbf{w}_1^2 = (0, 1)$.

Because of the simple structure of the set P_2, instead of introducing an extreme ray $\mathbf{w}_2^1$ and an associated variable θ_2^1, we identify θ_2^1 with x_3, and keep x_3 as a variable in the master problem.

The master problem has two equality constraints, namely,

$$\sum_{j=1}^{3} \lambda_1^j \mathbf{x}_1^j + \sum_{k=1}^{2} \theta_1^k \mathbf{w}_1^k + x_3 = 8,$$

$$\sum_{j=1}^{3} \lambda_1^j = 1.$$

Accordingly, a basis consists of exactly two columns.

In this example, we have $\mathbf{D}_1 = [1\ 0]$ and $\mathbf{D}_2 = 1$. Consider the variable λ_1^1 associated with the extreme point $\mathbf{x}_1^1 = (6, 2)$ of the first subproblem. The corresponding column is $(\mathbf{D}_1 \mathbf{x}_1^1, 1) = (6, 1)$. The column associated with x_3 is $(1, 0)$. If we choose λ_1^1 and x_3 as the basic variables, the basis matrix is

$$\mathbf{B} = \begin{bmatrix} 6 & 1 \\ 1 & 0 \end{bmatrix},$$

and its inverse is

$$\mathbf{B}^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & -6 \end{bmatrix}.$$

We form the product of $\mathbf{B}^{-1}$ with the vector $(\mathbf{b}_0, 1) = (8,1)$. The result, which is $(1,2)$, gives us the values of the basic variables λ_1^1, x_3. Since these values are nonnegative, we have a basic feasible solution to the master problem, which corresponds to $(x_1, x_2) = \mathbf{x}_1^1 = (6,2)$; see Figure 6.4.

We now determine the dual variables. We have $c_{B(1)} = (-5,1)'\mathbf{x}_1^1 = -28$ and $c_{B(2)} = 0 \times 1 = 0$. We therefore have

$$\mathbf{p}' = \begin{bmatrix} q & r_1 \end{bmatrix} = \mathbf{c}_B'\mathbf{B}^{-1} = \begin{bmatrix} -28 & 0 \end{bmatrix}\mathbf{B}^{-1} = \begin{bmatrix} 0 & -28 \end{bmatrix}.$$

(Note that we use the notation q instead of $\mathbf{q}$, because $\mathbf{q}$ is one-dimensional. Furthermore, the dual variable r_2 is absent because there is no convexity constraint associated with the second subproblem.)

We form the first subproblem. We are to minimize $(\mathbf{c}_1' - \mathbf{q}'\mathbf{D}_1)\mathbf{x}_1$ subject to $\mathbf{x}_1 \in P_1$. Because $q = 0$, we have $\mathbf{c}_1' - q\mathbf{D}_1 = \mathbf{c}_1' = (-5,1)'$. We are therefore minimizing $-5x_1 + x_2$ subject to $\mathbf{x}_1 \in P_1$ and the optimal cost is $-\infty$. In particular, we find that the extreme ray $\mathbf{w}_1^1 = (1,2)$ has negative cost. The associated variable θ_1^1 is to enter the basis. At this point, we generate the column corresponding to θ_1^1. Since $\mathbf{D}_1\mathbf{w}_1^1 = 1$, the corresponding column, call it $\mathbf{g}$, is $(1,0)$. We form the vector $\mathbf{u} = \mathbf{B}^{-1}\mathbf{g}$, which is found to be $(0,1)$. The only positive entry is the second one and this is therefore the pivot element. It follows that the second basic variable, namely x_3, exits the basis. Because the column associated with the entering variable θ_1^1 is equal to the column associated with the exiting variable x_3, we still have the same basis matrix and, therefore, the same values of the basic variables, namely, $\lambda_1^1 = 1$, $\theta_1^1 = 2$. This takes us to the vector $\mathbf{x} = \mathbf{x}_1^1 + 2\mathbf{w}_1^1 = (8,6)$; see Figure 6.4.

For the new basic variables, $c_{B(1)}$ is again -28 and

$$c_{B(2)} = \mathbf{c}_1'\mathbf{w}_1^1 = \begin{bmatrix} -5 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} = -3.$$

We compute $(q, r_1)' = \mathbf{c}_B'\mathbf{B}^{-1}$, which is equal to $(-3,-10)'$.

We now go back to the first subproblem. Since $q = -3$, we have

$$\mathbf{c}_1' - q\mathbf{D}_1 = \begin{bmatrix} -5 & 1 \end{bmatrix} - (-3)\begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} -2 & 1 \end{bmatrix}.$$

We minimize $-2x_1 + x_2$ over the set P_1. The optimal cost is -10 and is attained at $(x_1, x_2) = (8,6)$. Because the optimal cost -10 is equal to r_1, all of the variables associated with the first subproblem have nonnegative reduced costs.

We next consider the second subproblem. We have $\mathbf{c}_2 = 0$, $q = -3$, and $\mathbf{D}_2 = 1$. Thus, the reduced cost of x_3 is equal to $\mathbf{c}_2' - q\mathbf{D}_2 = 3$. We conclude that all of the variables in the master problem have nonnegative reduced costs and we have an optimal solution.

Starting the algorithm

In order to start the decomposition algorithm, we need to find a basic feasible solution to the master problem. This can be done as follows. We first apply Phase I of the simplex method to each one of the polyhedra P_1 and P_2, separately, and find extreme points $\mathbf{x}_1^1$ and $\mathbf{x}_2^1$ of P_1 and P_2, respectively. By possibly multiplying both sides of some of the coupling constraints by -1, we can assume that $\mathbf{D}_1\mathbf{x}_1^1 + \mathbf{D}_2\mathbf{x}_2^1 \le \mathbf{b}$. Let $\mathbf{y}$ be a vector of auxiliary variables of dimension m_0. We form the auxiliary master problem

$$\begin{aligned} \text{minimize} \quad & \sum_{t=1}^{m_0} y_t \\ \text{subject to} \quad & \sum_{i=1,2}\left(\sum_{j\in J_i}\lambda_i^j\mathbf{D}_i\mathbf{x}_i^j + \sum_{k\in K_i}\theta_i^k\mathbf{D}_i\mathbf{w}_i^k\right) + \mathbf{y} = \mathbf{b}_0 \\ & \sum_{j\in J_1}\lambda_1^j = 1 \\ & \sum_{j\in J_2}\lambda_2^j = 1 \\ & \lambda_i^j \ge 0,\ \theta_i^k \ge 0,\ y_t \ge 0, \qquad \forall\, i,j,k,t. \end{aligned}$$

A basic feasible solution to the auxiliary problem is obtained by letting $\lambda_1^1 = \lambda_2^1 = 1$, $\lambda_i^j = 0$ for $j \ne 1$, $\theta_i^k = 0$ for all k, and $\mathbf{y} = \mathbf{b}_0 - \mathbf{D}_1\mathbf{x}_1^1 - \mathbf{D}_2\mathbf{x}_2^1$. Starting from here, we can use the decomposition algorithm to solve the auxiliary master problem. If the optimal cost is positive, then the master problem is infeasible. If the optimal cost is zero, an optimal solution to the auxiliary problem provides us with a basic feasible solution to the master problem.

Termination and computational experience

The decomposition algorithm is a special case of the revised simplex method and, therefore, inherits its termination properties. In particular, in the absence of degeneracy, it is guaranteed to terminate in a finite number of steps. In the presence of degeneracy, finite termination is ensured if an anticycling rule is used, although this is rarely done in practice. Note that Bland's rule cannot be applied in this context, because it is incompatible with the way that the decomposition algorithm chooses the entering variable. There is no such difficulty, in principle, with the lexicographic pivoting rule, provided that the inverse basis matrix is explicitly computed.

A practical way of speeding up the solution of the subproblems is to start the simplex method on a subproblem using the optimal solution obtained the previous time that the subproblem was solved. As long as the objective function of the subproblem does not change too drastically between successive master iterations, one expects that this could lead to an optimal solution for the subproblem after a relatively small number of iterations.

Practical experience suggests that the algorithm makes substantial progress in the beginning, but the cost improvement can become very slow later on. For this reason, the algorithm is sometimes terminated prematurely, yielding a suboptimal solution.

The available experience also suggests that the algorithm is usually no faster than the revised simplex method applied to the original problem. The true advantage of the decomposition algorithm lies in its storage requirements. Suppose that we have t subproblems, each one having the same number m_1 of equality constraints. The storage requirements of the revised simplex method for the original problem are $O\big((m_0+tm_1)^2\big)$, which is the size of the revised simplex tableau. In contrast, the storage requirements of the decomposition algorithm are $O\big((m_0+t)^2\big)$ for the tableau of the master problem, and t times $O(m_1^2)$ for the revised simplex tableaux of the subproblems. Furthermore, the decomposition algorithm needs to have only one tableau stored in main memory at any given time. For example, if $t=10$ and if $m_0=m_1$ is much larger than t, the main memory requirements of the decomposition algorithm are about 100 times smaller than those of the revised simplex method. With memory being a key bottleneck in handling very large linear programming problems, the decomposition approach can substantially enlarge the range of problems that can be practically solved.

Bounds on the optimal cost

As already discussed, the decomposition algorithm may take a long time to terminate, especially for very large problems. We will now show how to obtain upper and lower bounds for the optimal cost. Such bounds can be used to stop the algorithm once the cost gets acceptably close to the optimum.

Theorem 6.1 *Suppose that the master problem is feasible and its optimal cost z^* is finite. Let z be the cost of the feasible solution obtained at some intermediate stage of the decomposition algorithm. Also, let r_i be the value of the dual variable associated with the convexity constraint for the ith subproblem. Finally, let z_i be the optimal cost in the ith subproblem, assumed finite. Then,*

$$z+\sum_i (z_i-r_i)\le z^*\le z.$$

Proof. The inequality $z^*\le z$ is obvious, since z is the cost associated with a feasible solution to the original problem. It remains to prove the left-hand side inequality in the statement of the theorem.

We provide the proof for the case of two subproblems. The proof for

the general case is similar. The dual of the master problem is

$$\begin{array}{lllll}
\text{maximize} & \mathbf{q}'\mathbf{b}_0+r_1+r_2 & & & \\
\text{subject to} & \mathbf{q}'\mathbf{D}_1\mathbf{x}_1^j + r_1 & \le & \mathbf{c}_1'\mathbf{x}_1^j, & \forall\, j\in J_1,\\
 & \mathbf{q}'\mathbf{D}_1\mathbf{w}_1^k & \le & \mathbf{c}_1'\mathbf{w}_1^k, & \forall\, k\in K_1,\\
 & \mathbf{q}'\mathbf{D}_2\mathbf{x}_2^j + r_2 & \le & \mathbf{c}_2'\mathbf{x}_2^j, & \forall\, j\in J_2,\\
 & \mathbf{q}'\mathbf{D}_2\mathbf{w}_2^k & \le & \mathbf{c}_2'\mathbf{w}_2^k, & \forall\, k\in K_2.
\end{array} \tag{6.14}$$

Suppose that we have a basic feasible solution to the master problem, with cost z, and let $(\mathbf{q},r_1,r_2)$ be the associated vector of simplex multipliers. This is a (generally infeasible) basic solution to the dual problem, with the same cost, that is,

$$\mathbf{q}'\mathbf{b}_0+r_1+r_2=z. \tag{6.15}$$

Since the optimal cost z_1 in the first subproblem is finite, we have

$$\begin{aligned}
\min_{j\in J_1}(\mathbf{c}_1'\mathbf{x}_1^j-\mathbf{q}'\mathbf{D}_1\mathbf{x}_1^j) &= z_1,\\
\min_{k\in K_1}(\mathbf{c}_1'\mathbf{w}_1^k-\mathbf{q}'\mathbf{D}_1\mathbf{w}_1^k) &\ge 0.
\end{aligned}$$

Thus, $\mathbf{q}$ together with z_1 in the place of r_1, satisfy the first two dual constraints. By a similar argument, $\mathbf{q}$ together with z_2 in the place of r_2, satisfy the last two dual constraints. Therefore, $(\mathbf{q},z_1,z_2)$ is a feasible solution to the dual problem (6.14). Its cost is $\mathbf{q}'\mathbf{b}_0+z_1+z_2$ and, by weak duality, is no larger than the optimal cost z^*. Hence,

$$\begin{aligned}
z^* &\ge \mathbf{q}'\mathbf{b}_0+z_1+z_2\\
&= \mathbf{q}'\mathbf{b}_0+r_1+r_2+(z_1-r_1)+(z_2-r_2)\\
&= z+(z_1-r_1)+(z_2-r_2),
\end{aligned}$$

where the last equality follows from Eq. (6.15). □

Note that if the optimal cost in one of the subproblems is $-\infty$, then Theorem 6.1 does not provide any useful bounds.

The proof of Theorem 6.1 is an instance of a general method for obtaining lower bounds on the optimal cost of linear programming problems, which is the following. Given a nonoptimal basic feasible solution to the primal, we consider the corresponding (infeasible) basic solution to the dual problem. If we can somehow modify this dual solution, to make it feasible, the weak duality theorem readily yields a lower bound. This was the approach taken in the proof of Theorem 6.1, where we started from the generally infeasible dual solution $(\mathbf{q},r_1,r_2)$, moved to the dual feasible solution $(\mathbf{q},z_1,z_2)$, and then invoked weak duality.

Example 6.4 Let us revisit Example 6.2 and consider the situation just before the first change of basis. We are at a basic feasible solution determined by

$(\lambda^1, \lambda^2) = (0.8, 0.2)$. Since $\mathbf{c}_B = (-22, -17)$, we have $z = (-22, -17)'(0.8, 0.2) = -21$. We also have $r = -4$. Finally, the optimal cost in the subproblem is $(-1, 1, -2)'(2, 1, 2) = -5$. It follows that $-21 \geq z^* \geq -21 + (-5) - (-4) = -22$. Indeed, the true value of z^* is -21.5.

6.5 Stochastic programming and Benders decomposition

In this section, we introduce and study two-stage stochastic linear programming problems. In this important class of problems, there are two sets of decisions that are made in consecutive stages. Furthermore, there are some exogenous parameters that influence the second stage of decision making, but whose values are uncertain, and only become known after the first set of decisions has been fixed. In order to address problems of this type, we develop a new decomposition algorithm, called Benders decomposition, which is based on delayed constraint generation (as opposed to delayed column generation).

Problem formulation

Consider a decision maker who has to act in two consecutive stages. The first stage involves the choice of a decision vector $\mathbf{x}$. Subsequently, some new information is obtained, and then, at the second stage, a new vector $\mathbf{y}$ of decisions is to be chosen. Regarding the nature of the obtained information, we assume that there are K possible scenarios, and that the true scenario is only revealed after $\mathbf{x}$ is chosen. We use ω to index the different scenarios, and we let α_ω stand for the probability of any particular scenario ω, which we assume to be positive. Since the second stage decisions are made after the true scenario ω becomes known, we allow the decision vector $\mathbf{y}$ to depend on ω, and we use the notation $\mathbf{y}_\omega$ to make this dependence explicit.

We now make more specific assumptions on the problem objectives and constraints. We are given cost vectors $\mathbf{c}$ and $\mathbf{f}$, associated with the decisions $\mathbf{x}$ and $\mathbf{y}_\omega$, respectively. The first stage decisions must satisfy the constraints

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0}. \end{aligned}$$

In addition, the first and second stage decisions need to satisfy constraints of the form

$$\begin{aligned} \mathbf{B}_\omega \mathbf{x} + \mathbf{D}\mathbf{y}_\omega &= \mathbf{d}_\omega \\ \mathbf{y}_\omega &\geq \mathbf{0}, \end{aligned}$$

for all ω; in particular, every scenario may involve a different set of constraints. The objective is to choose $\mathbf{x}$ and $\mathbf{y}_1, \ldots, \mathbf{y}_K$ so that the "expected cost"

$$\mathbf{c}'\mathbf{x} + \alpha_1 \mathbf{f}'\mathbf{y}_1 + \cdots + \alpha_K \mathbf{f}'\mathbf{y}_K$$

is minimized. We thus arrive at the problem

$$\begin{array}{lllllll} \text{minimize} & \mathbf{c}'\mathbf{x} & + \ \alpha_1\mathbf{f}'\mathbf{y}_1 & + \ \alpha_2\mathbf{f}'\mathbf{y}_2 & + \ \cdots & + \ \alpha_K\mathbf{f}'\mathbf{y}_K & \\ \text{subject to} & \mathbf{A}\mathbf{x} & & & & & = \mathbf{b} \\ & \mathbf{B}_1\mathbf{x} & + \ \mathbf{D}\mathbf{y}_1 & & & & = \mathbf{d}_1 \\ & \mathbf{B}_2\mathbf{x} & & + \ \mathbf{D}\mathbf{y}_2 & & & = \mathbf{d}_2 \\ & \vdots & & & \ddots & & \vdots \\ & \mathbf{B}_K\mathbf{x} & & & & + \ \mathbf{D}\mathbf{y}_K & = \mathbf{d}_K \\ & \multicolumn{6}{l}{\quad \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K \geq \mathbf{0},} \end{array}$$

which will be referred to as the *master problem.* Notice that even if the number of possible scenarios K is moderate, this formulation can be a large linear programming problem. For this reason, a decomposition method is in order.

Example 6.5 (Electric power capacity expansion) An electric utility is installing two generators (indexed by $j = 1, 2$) with different fixed and operating costs, in order to meet the demand within its service region. Each day is divided into three parts of equal duration, indexed by $i = 1, 2, 3$. These correspond to parts of the day during which demand takes a base, medium, or peak value, respectively. The fixed cost per unit capacity of generator j is amortized over its lifetime and amounts to c_j per day. The operating cost of generator j during the ith part of the day is f_{ij}. If the demand during the ith part of the day cannot be served due to lack of capacity, additional capacity must be purchased at a cost of g_i. Finally, the capacity of each generator j is required to be at least b_j.

There are two sources of uncertainty, namely, the exact value of the demand d_i during each part of the day, and the availability a_j of generator j. The demand d_i can take one of four values $d_{i,1}, \ldots, d_{i,4}$, with probability $p_{i,1}, \ldots, p_{i,4}$, respectively. The availability of generator 1 is $a_{1,1}, \ldots, a_{1,4}$, with probability $q_{1,1}, \ldots, q_{1,4}$, respectively. Similarly, the availability of generator 2 is $a_{2,1}, \ldots, a_{2,5}$, with probability $q_{2,1}, \ldots, q_{2,5}$, respectively. If we enumerate all the possible events, we see that there is a total of $4^3 \times 4 \times 5 = 1280$ scenarios ω. Let us use d_i^ω and a_j^ω to denote the demands and availabilities, respectively, under scenario ω.

We introduce the first stage variables x_j, $j = 1, 2$, that represent the installed capacity of generator j. We also introduce the second stage variables y_{ij}^ω that denote the operating levels of generator j during the ith part of the day and under scenario ω. Finally y_i^ω is the capacity that needs to be purchased under scenario ω, in order to satisfy unmet demand during the ith part of the day. We interpret availability to mean that the operating level of generator j, at any given time, is at most $a_j x_j$. We then arrive at the following formulation:

$$\begin{aligned}
\text{minimize} \quad & \sum_{j=1}^{2} c_j x_j + E\left[\sum_{i=1}^{3}\left(\sum_{j=1}^{2} f_{ij} y_{ij}^{\omega} + g_i y_i^{\omega}\right)\right] \\
\text{subject to} \quad & x_j \ge b_j, && \forall\, j, \\
& y_{ij}^{\omega} \le a_j^{\omega} x_j, && \forall\, i, j, \omega, \\
& \sum_{j=1}^{2} y_{ij}^{\omega} + y_i^{\omega} \ge d_i^{\omega}, && \forall\, i, \omega, \\
& x_j, y_{ij}^{\omega}, y_i^{\omega} \ge 0, && \forall\, i, j, \omega.
\end{aligned}$$

(Here, $E[\cdot]$ stands for mathematical expectation, that is the average over all scenarios ω, weighted according to their probabilities.) The full model involves 11522 variables and 11522 constraints (not counting nonnegativity constraints).

Reformulation of the problem

Consider a vector $\mathbf{x}$ such that $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x} \ge \mathbf{0}$, and suppose that this is our choice for the first stage decisions. Once $\mathbf{x}$ is fixed, the optimal second stage decisions $\mathbf{y}_\omega$ can be determined separately from each other, by solving for each ω the problem

$$\begin{aligned}
\text{minimize} \quad & \mathbf{f}'\mathbf{y}_\omega \\
\text{subject to} \quad & \mathbf{B}_\omega \mathbf{x} + \mathbf{D}\mathbf{y}_\omega = \mathbf{d}_\omega \\
& \mathbf{y}_\omega \ge \mathbf{0}.
\end{aligned} \tag{6.16}$$

Let $z_\omega(\mathbf{x})$ be the optimal cost of the problem (6.16), together with the convention $z_\omega(\mathbf{x}) = \infty$ if the problem is infeasible. If we now go back to the optimization of $\mathbf{x}$, we are faced with the problem

$$\begin{aligned}
\text{minimize} \quad & \mathbf{c}'\mathbf{x} + \sum_{\omega=1}^{K} \alpha_\omega z_\omega(\mathbf{x}) \\
\text{subject to} \quad & \mathbf{Ax} = \mathbf{b} \\
& \mathbf{x} \ge \mathbf{0}.
\end{aligned} \tag{6.17}$$

Of course, in solving this problem, we should only consider those $\mathbf{x}$ for which none of the $z_\omega(\mathbf{x})$ are equal to infinity.

We will approach problem (6.16) by forming its dual, which is

$$\begin{aligned}
\text{maximize} \quad & \mathbf{p}_\omega'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}) \\
\text{subject to} \quad & \mathbf{p}_\omega'\mathbf{D} \le \mathbf{f}'.
\end{aligned} \tag{6.18}$$

Let

$$P = \{\mathbf{p} \mid \mathbf{p}'\mathbf{D} \le \mathbf{f}'\}.$$

We assume that P is nonempty and has at least one extreme point. Let $\mathbf{p}^i$, $i = 1, \ldots, I$, be the extreme points, and let $\mathbf{w}^j$, $j = 1, \ldots, J$, be a complete set of extreme rays of P.

Under our assumption that the set P is nonempty, either the dual problem (6.18) has an optimal solution and $z_\omega(\mathbf{x})$ is finite, or the optimal dual cost is infinite, the primal problem (6.16) is infeasible, and $z_\omega(\mathbf{x}) = \infty$. In particular, $z_\omega(\mathbf{x}) < \infty$ if and only if

$$(\mathbf{w}^j)'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}) \le 0, \qquad \forall\, j. \tag{6.19}$$

Whenever $z_\omega(\mathbf{x})$ is finite, it is the optimal cost of the problem (6.18), and the optimum must be attained at an extreme point of the set P; in particular,

$$z_\omega(\mathbf{x}) = \max_{i=1,\ldots,I} (\mathbf{p}^i)'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}).$$

Alternatively, $z_\omega(\mathbf{x})$ is the smallest number z_ω such that

$$(\mathbf{p}^i)'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}) \le z_\omega, \qquad \forall\, i.$$

We use this characterization of $z_\omega(\mathbf{x})$ in the original problem (6.17), and also take into account the condition (6.19), which is required for $z_\omega(\mathbf{x})$ to be finite, and we conclude that the master problem (6.17) can be put in the form

$$\begin{aligned}
\text{minimize} \quad & \mathbf{c}'\mathbf{x} + \sum_{\omega=1}^{K} \alpha_\omega z_\omega \\
\text{subject to} \quad & \mathbf{Ax} = \mathbf{b} \\
& (\mathbf{p}^i)'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}) \le z_\omega, && \forall\, i, \omega, \\
& (\mathbf{w}^j)'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}) \le 0, && \forall\, j, \omega, \\
& \mathbf{x} \ge \mathbf{0}.
\end{aligned} \tag{6.20}$$

With this reformulation, the number of variables has been reduced substantially. The number of constraints can be extremely large, but this obstacle can be overcome using the cutting plane method. In particular, we will only generate constraints that we find to be violated by the current solution.

Delayed constraint generation

At a typical iteration of the algorithm, we consider the *relaxed master problem*, which has the same objective as the problem (6.20), but involves only a subset of the constraints. We assume that we have an optimal solution to the relaxed master problem, consisting of a vector $\mathbf{x}^*$ and a vector $\mathbf{z}^* = (z_1^*, \ldots, z_K^*)$. In the spirit of the cutting plane method, we need to check whether $(\mathbf{x}^*, \mathbf{z}^*)$ is also a feasible solution to the full master problem. However, instead of individually checking all of the constraints, we proceed by solving some auxiliary subproblems.

For $\omega = 1, \ldots, K$, we consider the subproblem

$$\begin{aligned} \text{minimize} \quad & \mathbf{f}'\mathbf{y}_\omega \\ \text{subject to} \quad & \mathbf{D}\mathbf{y}_\omega = \mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}^* \\ & \mathbf{y}_\omega \geq \mathbf{0}, \end{aligned}$$

which we solve to optimality. Notice that the subproblems encountered at different iterations, or for different values of ω, differ only in the right-hand side vector $\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}^*$. In particular, the corresponding dual problems always have the same feasible set, namely, P. For this reason, it is natural to assume that the subproblems are solved by means of the dual simplex method.

There are a few different possibilities to consider:

(a) If the dual simplex method indicates that a (primal) subproblem is infeasible, it provides us with an extreme ray $\mathbf{w}^{j(\omega)}$ of the dual feasible set P, such that

$$(\mathbf{w}^{j(\omega)})'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}^*) > 0.$$

We have then identified a violated constraint, which can be added to the relaxed master problem.

(b) If a primal subproblem is feasible, then the dual simplex method terminates, and provides us with a dual optimal basic feasible solution $\mathbf{p}^{i(\omega)}$. If we have

$$(\mathbf{p}^{i(\omega)})'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}^*) > z_\omega^*,$$

we have again identified a violated constraint, which can be added to the relaxed master problem.

(c) Finally, if the primal subproblems are all feasible and we have

$$(\mathbf{p}^{i(\omega)})'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}^*) \leq z_\omega^*,$$

for all ω, then, by the optimality of $\mathbf{p}^{i(\omega)}$, we obtain

$$(\mathbf{p}^{i})'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}^*) \leq z_\omega^*,$$

for all extreme points $\mathbf{p}^i$. In addition,

$$(\mathbf{w}^{j})'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}^*) \leq 0,$$

for all extreme rays $\mathbf{w}^j$, and no constraint is violated. We then have an optimal solution to the master problem (6.20), and the algorithm terminates.

The resulting algorithm is summarized below.

Benders decomposition for two-stage problems

1. A typical iteration starts with a relaxed master problem, in which only some of the constraints of the master problem (6.20) are included. An optimal solution $(\mathbf{x}^*, \mathbf{z}^*)$ to the relaxed master problem is calculated.
2. For every ω, we solve the subproblem

$$\begin{aligned} \text{minimize} \quad & \mathbf{f}'\mathbf{y}_\omega \\ \text{subject to} \quad & \mathbf{D}\mathbf{y}_\omega = \mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}^* \\ & \mathbf{y}_\omega \geq \mathbf{0}, \end{aligned}$$

   using the dual simplex method.
3. If for every ω, the corresponding subproblem is feasible and the optimal cost is less than or equal to z_ω^*, all constraints are satisfied, we have an optimal solution to the master problem, and the algorithm terminates.
4. If the subproblem corresponding to some ω has an optimal solution whose cost is greater than z_ω^*, an optimal basic feasible solution $\mathbf{p}^{i(\omega)}$ to the dual of the subproblem is identified, and the constraint

$$(\mathbf{p}^{i(\omega)})'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}) \leq z_\omega$$

   is added to the relaxed master problem.
5. If the subproblem corresponding to some ω is infeasible, its dual has infinite cost, and a positive cost extreme ray $\mathbf{w}^{j(\omega)}$ is identified. Then, the constraint

$$(\mathbf{w}^{j(\omega)})'(\mathbf{d}_\omega - \mathbf{B}_\omega \mathbf{x}) \leq 0$$

   is added to the relaxed master problem.

Benders decomposition uses delayed constraint generation and the cutting plane method, and should be contrasted with Dantzig-Wolfe decomposition, which is based on column generation. Nevertheless, the two methods are almost identical, with Benders decomposition being essentially the same as Dantzig-Wolfe decomposition applied to the dual. Let us also note, consistently with our discussion in Section 6.3, that we have the option of discarding all or some of the constraints in the relaxed primal that have become inactive.

One of the principal practical difficulties with stochastic programming, is that the number K of possible scenarios is often large, leading to a large number of subproblems. This is even more so for stochastic programming problems involving more than two stages, where similar methods can be in principle applied. A number of remedies have been proposed, including the use of random sampling to generate only a representative set of scenarios. With the use of parallel computers and sophisticated sampling methods, the solution of some extremely large problems may become possible.

6.6 Summary

The main ideas developed in this chapter are the following:

(a) In a problem with an excessive number of columns, we need to generate a column only if its reduced cost is negative, and that column is to enter the basis (delayed column generation). A method of this type requires an efficient subroutine for identifying a variable with negative reduced cost.

(b) In a problem with an excessive number of constraints, a constraint needs to be taken into account only if it is violated by the current solution (delayed constraint generation). A method of this type (cutting plane method) requires an efficient subroutine for identifying violated constraints.

We have noted that delayed column generation methods applied to the primal coincide with cutting plane methods applied to the dual. Furthermore, we noted that there are several variants depending on whether we retain or discard from memory previously generated columns or constraints.

For a problem consisting of a number of subproblems linked by coupling constraints, the delayed column generation method applied to a suitably reformulated problem, results in the Dantzig-Wolfe decomposition method. Loosely speaking, Benders decomposition is the "dual" of Dantzig-Wolfe decomposition, and is based on delayed constraint generation. Stochastic programming is an important class of problems where Benders decomposition can be applied.

6.7 Exercises

Exercise 6.1 Consider the cutting stock problem. Use an optimal solution to the linear programming problem (6.4) to construct a feasible solution for the corresponding integer programming problem, whose cost differs from the optimal cost by no more than m.

Exercise 6.2 This problem is a variation of the diet problem. There are n foods and m nutrients. We are given an $m \times n$ matrix $\mathbf{A}$, with a_{ij} specifying the amount of nutrient i per unit of the jth food. Consider a parent with two children. Let $\mathbf{b}^1$ and $\mathbf{b}^2$ be the minimal nutritional requirements of the two children, respectively. Finally, let $\mathbf{c}$ be the cost vector with the prices of the different foods. Assume that $a_{ij} \geq 0$ and $c_i > 0$ for all i and j.

The parent has to buy food to satisfy the children's needs, at minimum cost. To avoid jealousy, there is the additional constraint that the amount to be spent for each child is the same.

(a) Provide a standard form formulation of this problem. What are the dimensions of the constraint matrix?

(b) If the Dantzig-Wolfe method is used to solve the problem in part (a), construct the subproblems solved during a typical iteration of the master problem.

(c) Suggest a direct approach for solving this problem based on the solution of two single-child diet problems.

Exercise 6.3 Consider the following linear programming problem:

$$\begin{array}{llcl}
\text{maximize} & x_{12} + x_{22} + x_{23} & & \\
\text{subject to} & x_{11} + x_{12} + x_{13} & = & 20 \\
& x_{21} + x_{22} + x_{23} & = & 20 \\
& -x_{11} - x_{21} & = & -20 \\
& -x_{12} - x_{22} & = & -10 \\
& -x_{13} - x_{23} & = & -10 \\
& x_{11} + x_{23} & \leq & 15 \\
& x_{ij} \geq 0, \quad \text{for all } i, j. & &
\end{array}$$

We wish to solve this problem using Dantzig-Wolfe decomposition, where the constraint $x_{11} + x_{23} \leq 15$ is the only "coupling" constraint and the remaining constraints define a single subproblem.

(a) Consider the following two feasible solutions for the subproblem:

$$\mathbf{x}^1 = (x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}) = (20, 0, 0, 0, 10, 10),$$

and

$$\mathbf{x}^2 = (x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}) = (0, 10, 10, 20, 0, 0).$$

Construct a restricted master problem in which $\mathbf{x}$ is constrained to be a convex combination of $\mathbf{x}^1$ and $\mathbf{x}^2$. Find the optimal solution and the optimal simplex multipliers for the restricted master problem.

(b) Using the simplex multipliers calculated in part (a), formulate the subproblem and solve it by inspection.

(c) What is the reduced cost of the variable λ_i associated with the optimal extreme point $\mathbf{x}^i$ obtained from the subproblem solved in part (b)?

(d) Compute an upper bound on the optimal cost.

Exercise 6.4 Consider a linear programming problem of the form

$$\begin{array}{ll}
\text{minimize} & \mathbf{c}_1'\mathbf{x}_1 + \mathbf{c}_2'\mathbf{x}_2 + \mathbf{c}_0'\mathbf{y} \\
\text{subject to} & \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} & \mathbf{F}_1 \\ \mathbf{0} & \mathbf{D}_2 & \mathbf{F}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{y} \end{bmatrix} \geq \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \\
& \mathbf{x}_1, \mathbf{x}_2 \geq \mathbf{0}.
\end{array}$$

We will develop two different ways of decomposing this problem.

(a) Form the dual problem and explain how Dantzig-Wolfe decomposition can be applied to it. What is the structure of the subproblems solved during a typical iteration?

(b) Rewrite the first set of constraints in the form $\mathbf{D}_1\mathbf{x}_1 + \mathbf{F}_1\mathbf{y}_1 \geq \mathbf{b}_1$ and $\mathbf{D}_2\mathbf{x}_2 + \mathbf{F}_2\mathbf{y}_2 \geq \mathbf{b}_2$, together with a constraint relating $\mathbf{y}_1$ to $\mathbf{y}_2$. Discuss how to apply Dantzig-Wolfe decomposition and describe the structure of the subproblems solved during a typical iteration.

Exercise 6.5 Consider a linear programming problem of the form

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}'\mathbf{x} + \mathbf{d}'\mathbf{y} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + \mathbf{D}\mathbf{y} \leq \mathbf{b} \\ & \mathbf{F}\mathbf{x} \leq \mathbf{f} \\ & \mathbf{y} \geq \mathbf{0}. \end{aligned}$$

(a) Suppose that we have access to a very fast subroutine for solving problems of the form

$$\begin{aligned} \text{minimize} \quad & \mathbf{h}'\mathbf{x} \\ \text{subject to} \quad & \mathbf{F}\mathbf{x} \leq \mathbf{f}, \end{aligned}$$

for arbitrary cost vectors $\mathbf{h}$. How would you go about decomposing the problem?

(b) Suppose that we have access to a very fast subroutine for solving problems of the form

$$\begin{aligned} \text{minimize} \quad & \mathbf{d}'\mathbf{y} \\ \text{subject to} \quad & \mathbf{D}\mathbf{y} \leq \mathbf{h} \\ & \mathbf{y} \geq \mathbf{0}, \end{aligned}$$

for arbitrary right-hand side vectors $\mathbf{h}$. How would you go about decomposing the problem?

Exercise 6.6 Consider a linear programming problem in standard form in which the matrix $\mathbf{A}$ has the following structure:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{00} & \mathbf{A}_{01} & \cdots & \cdots & \mathbf{A}_{0n} \\ \mathbf{A}_{10} & \mathbf{A}_{11} & & & \\ \vdots & & \mathbf{A}_{22} & & \\ \vdots & & & \ddots & \\ \mathbf{A}_{n0} & & & & \mathbf{A}_{nn} \end{bmatrix}.$$

(All submatrices other than those indicated are zero.) Show how a decomposition method can be applied to a problem with this structure. Do not provide details, as long as you clearly indicate the master problem and the subproblems. *Hint:* Decompose twice.

Exercise 6.7 Consider a linear programming problem in standard form. Let us treat the equality constraints as the "coupling" constraints and use the Dantzig-Wolfe decomposition method, for the case of a single subproblem. Show that the resulting master problem is identical to the problem that we started with.

Exercise 6.8 Consider the Dantzig-Wolfe decomposition method and suppose that we are at a basic feasible solution to the master problem.

(a) Show that at least one of the variables λ_1^j must be a basic variable.

(b) Let r_1 be the current value of the simplex multiplier associated with the first convexity constraint (6.12), and let z_1 be the optimal cost in the first subproblem. Show that $z_1 \leq r_1$.

Exercise 6.9 Consider a problem of the form

$$\text{minimize} \quad \max_{i=1,\ldots,m} (\mathbf{a}_i'\mathbf{x} - b_i),$$

subject to no constraints, where $\mathbf{a}_i$, b_i are given vectors and scalars, respectively.

(a) Describe a cutting plane method for problems of this form.

(b) Let $\mathbf{x}$ be an optimal solution to a relaxed problem in which only some of the terms $\mathbf{a}_i'\mathbf{x} - b_i$ have been retained. Describe a simple method for obtaining lower and upper bounds on the optimal cost in the original problem.

Exercise 6.10 In this exercise, we develop an alternative proof of Theorem 6.1.

(a) Suppose that $\mathbf{x}$ is a basic feasible solution to a standard form problem, and let $\overline{\mathbf{c}}$ be the corresponding vector of reduced costs. Let $\mathbf{y}$ be any other feasible solution. Show that $\mathbf{c}'\mathbf{y} = \overline{\mathbf{c}}'\mathbf{y} + \mathbf{c}'\mathbf{x}$.

(b) Consider a basic feasible solution to the master problem whose cost is equal to z. Write down a lower bound on the reduced cost of any variable λ_i^j and θ_i^k, in terms of r_i and z_i. Then, use the result of part (a) to provide a proof of Theorem 6.1.

Exercise 6.11 (The relation between Dantzig-Wolfe and Benders decomposition) Consider the two-stage stochastic linear programming problem treated in Section 6.5.

(a) Show that the dual problem has a form which is amenable to Dantzig-Wolfe decomposition.

(b) Describe the Dantzig-Wolfe decomposition algorithm, as applied to the dual, and identify differences and similarities with Benders decomposition.

Exercise 6.12 (Bounds in Benders decomposition) For the two-stage stochastic linear programming problem of Section 6.5, derive upper and lower bounds on the optimal cost of the master problem, based on the information provided by the solutions to the subproblems.

6.8 Notes and sources

6.2. The delayed column generation approach to the cutting stock problem was put forth by Gilmore and Gomory (1961, 1963).

6.3. Cutting plane methods are often employed in linear programming approaches to integer programming problems, and will be discussed in Section 11.1. The same idea can also be applied to more general convex optimization problems; see, e.g., Bertsekas (1995b).

6.4. Dantzig-Wolfe decomposition was developed by Dantzig and Wolfe (1960). Example 6.2 is adapted from Bradley, Hax, and Magnanti (1977).

6.5. Stochastic programming began with work by Dantzig in the 1950's and has been extensively studied since then. Some books on this subject are Kall and Wallace (1994), and Infanger (1993); Example 6.5 is adapted from the latter reference. The Benders decomposition method was developed by Benders (1962). It finds applications in other contexts as well, such as discrete optimization; see, e.g., Schrijver (1986), and Nemhauser and Wosley (1988).

Chapter 7

Network flow problems

Contents